A Geometric Model-based Approach to Hand Gesture Recognition

Alexandre Calado*, Paolo Roselli†, Vito Errico*, Nathan Magrofuoco‡, Jean Vanderdonckt‡, and Giovanni Saggio*
*Department of Electronic Engineering, University of Rome "Tor Vergata", 00133 Rome, Italy
†Department of Mathematics, University of Rome "Tor Vergata", 00133 Rome, Italy
‡LouRIM, Université catholique de Louvain, B-1348 Louvain-la-Neuve, Belgium

***Abstract*—Arm-and-hand tracking by technological means allows gathering data that can be elaborated for determining gesture meaning. To this aim, machine learning algorithms have been mostly investigated looking for a balance between the highest recognition rate and the lowest recognition time. However, this balance comes mainly from statistical models, which are challenging to interpret. In contrast, we present μC^1 and μC^2, two geometric model-based approaches to gesture recognition which support the visualisation and geometrical interpretation of the recognition process. We compare μC^1 and μC^2 with respect to two classical machine learning algorithms, k-NN and SVM, and two state-of-the-art deep learning models, BiLSTM and GRU, on an experimental dataset of ten gesture classes from the Italian Sign Language (LIS), each repeated 100 times by five inexperienced non-native signers, and gathered with wearable technology (a sensory glove and inertial measurement units). As a result, we achieve a compromise between high recognition rates ($> 90\%$) and low recognition times ($< 0.1 sec$) that is adequate for human-computer interaction. Moreover, we elaborate on the algorithms' geometric interpretation based on geometric algebra, which supports some understanding of the recognition process.**

***Index Terms*—Geometric algebra, Gesture Recognition, Explainable Artificial Intelligence, Deep Learning, Machine Learning, Nearest Neighbour Classification.**

I. Introduction

GESTURES represent an effective way of user-interaction with machines, such as when controlling digital devices and musical instruments [1] or interacting within virtual environments. Moreover, gestures allow different types of manipulation, such as interacting with pets or, more intriguingly, performing surgical operations [2], [3]. Advantageously, gestures enable disabled people to gain autonomy by doing things through smart systems that convert gestures into actions [4]. Voluntary gestures are so important that when they are involuntarily performed, they can provide an objective "index" of a physiological or pathological issue [5].

Basically, gestures represent an effective and innate form of nonverbal communication useful in a noisy environment [6], in conveying non-semantic information [7], in emphasizing concepts [8] or moods [9] during speech and, above all, in providing a way of communication among deaf-mutes [10]. In fact, for people who do not have the possibility of being able to express themselves verbally, gestures represent the only means of communication. Advantageously, the automatic interpretation of gestures - with the aid of measurement devices and algorithms that translate the acquired data into verbal meaning - can be a strategic tool by allowing deaf-mutes to communicate with everyone, removing them from social "isolation". Indeed, this constitutes an intriguing research topic, particularly addressed by means of wearable technologies [11], [12] and Machine Learning (ML) [13].

Wearable technologies rely on devices for measuring electromyographic signals, spatial arrangement of the arm-and-hand, flexion of the arm-hand joints, and rotation of the wrist. On the other hand, ML provides many different classifiers, such as Support Vector Machines (SVM) and k-Nearest Neighbor (k-NN). Year after year, ML has meaningfully increased classification accuracy. However, the popularization of Deep Learning (DL) brought gesture recognition to a new level by outperforming previously used methods [14].

Although very high recognition rates can be achieved, DL and ML do not foster models that are graspable, lacking in interpretability and explainability, as they apply statistical methods that can be hard to understand. All in all, much of the State-of-the-Art (SotA) models can be viewed as "black boxes", where the complex logic supporting the classification is embedded in the method, leaving little or no space for understanding. To address this shortcoming, eXplainable Artificial Intelligence (XAI) benefits from increasing interest [15] (including in HCI [16]) by producing explainable models, interpretable to human users to increase the trust in AI discernment. This trend thus promotes "white boxes", where some interpretation of why recognition works is feasible.

Accordingly, we introduce μC^1 and μC^2, two gesture recognition geometric-based models partially inspired by Clifford Algebra and 2D stroke gesture recognition [17]. These models rely on Nearest Neighbor Classification (NNC), which consists in recognizing a candidate gesture by computing a distance between the latter and a finite number of reference gestures, named templates, contained in a training set. Due to its relatively simple implementation, NNC algorithms are computationally efficient, leading to low execution time and low memory consumption [18]. Since the algorithm's decision logic remains unchanged, training is not required when a reference gesture is added, removed, or modified in the training set, which is not the case for other classifiers. This allows the integration new gestures in the training set by the end-user.

Thus, we evaluate and compare μC^1 and μC^2 with respect

Manuscript received June 23, 2021; revised October 4, 2021. Corresponding author: A. Calado (email: Alexandre Calado alexandre-luis.airosocalado@students.uniroma2.eu).

to two classical machine learning methods - k-NN (k-Nearest Neighbors) and SVM (Support Vector Machine) - and two recurrent neural network (RNN) models based on different architectures - bidirectional long short term memory (BiLSTM) and gated recurrent unit (GRU), both SotA methods for gesture recognition - , on a sign language dataset acquired with a wearable sensor-based system. In contrast to other types of gestures used in HCI [19], sign languages are structured forms of hand-and-arm gestures that are used as a communication system and must be learned like any spoken language [20]. They are versatile but hard to learn and use because of their inherent dynamic complexity [21], which makes its automatic recognition challenging, thus representing an interesting test bench to assess our algorithms' robustness.

Since μC^1 and μC^2 are based on geometric algebra and the gesture data gathered by each sensor can be represented as sets of points in the 2 or 3-dimensional space, the recognition process can be visualized. Moreover, the geometric interpretation allows the algorithm's decision to be more intuitive and intelligible to humans, an aspect that we explore in this work.

II. Related Work

In the literature, different works report gesture recognition through NNC. For instance, k-NN has been extensively used [22], [23], including in Sign Language Recognition (SLR) tasks [13], [24], often combined with ML techniques to encode the gesture and ignoring gesture decomposition.

Nowadays, the use of 3D tracking devices, body-skeleton trackers and sensory gloves for HCI is more and more prevalent [25]. Since these devices can track the location of different points of interest in the body in real-time, a single gesture can be decomposed into multiple trajectories, which can be represented in the 2 or 3-dimensional space. So, by processing each of these trajectories, a simpler and more explainable alternative to complex ML techniques may be attained. For instance, NNC approaches used in 2-dimensional gesture stroke recognition can be extended for recognizing trajectories in the 3-dimensional domain. Accordingly, the popular \$-family of stroke gesture recognizers - which include the \$1 [26], \$N [27], \$P [28] and \$Q [29] recognizers – has been adapted to 3D gesture recognition in several works. In general, these algorithms are based on NNC by measuring the pairwise (dis)similarity of gesture shapes, and do not require large training sets to attain high recognition rates [30]. Besides, its relatively simple implementation allows rapid prototyping.

\$1 is a 2D single-stroke gesture recognizer based on four steps: resampling, rotation, scaling, and translation. First, the trajectory points are resampled to an isometric set of points to make gestures with different movement speeds comparable. To achieve rotation invariance, the trajectory is then rotated so the angle between its centroid and its first point is at 0°. Next, the trajectory is scaled non-uniformly to a reference square and it is translated so that its centroid is at (0,0). Finally, the similarity between two gestures is computed as the average Euclidean distance between corresponding points. The \$1 was perhaps the first 2D gesture stroke recognizer to be extended to the 3D domain with \$3 [31], [32], for which the authors used the 3D accelerometer data of a Nintendo Wii controller to classify 10 gestures with a recognition rate of 80%.

The Protractor [33] uses a pre-processing similar to \$1, however it differs in handling the orientation sensitivity and scaling, which preserves the aspect ratio of the gesture. Besides, rather than the Euclidean distance, it is used the angular cosine to compute the similarity between gestures. This algorithm was extended to 3D with Protractor3D [34], which presented a solution to the rotation invariance in 3D gestures by finding a rotation that minimizes the sum of squares error between the candidate and reference gestures. With only 5 reference gestures per class, a dataset of 3D accelerometer values was classified into 11 gestures, with a recognition rate ranging from 83.3% to 98.9%.

One Cent [35] took a step further in terms of simplicity by applying isometric resampling and transforming the gesture into a 1-dimensional vector, given by the Euclidean distance from the trajectory's centroid to each resampled point. Three Cent [30] explores this approach for recognizing 3D gestures on a public dataset, which achieved an 89.52% recognition rate when used together with orientation and finger descriptors, outperforming other StoA approaches.

$\$P^3$ and $\$Q^3$ [31], 3D extensions of \$P and \$Q, respectively, were compared with other algorithms that recognize 3D trajectories as 2D multi-stroke gestures: Rubine3D and Rubine-Sheng (3D extensions of Rubine [36]), \$F (a variant of $\$P^3$), FreeHandUni (derived from the Free-Hand pseudocode [37]) and $\$P{+}^3$ (a 3D extension of \$P+ [29]). These were evaluated in terms of recognition rate and execution time on three publicly available datasets, under both user-dependent and user-independent scenarios and with a reduced number of reference gestures (1 to 16).

Another stroke gesture recognizer, !FTL [18], aims to remove normalization, resampling, and rotation from the pre-processing step by converting the trajectory point to a series of pairs of vectors called "shapes", a concept that partially inspired our algorithms. The pairwise dissimilarity between two gestures is thereby given by the Local Shape Distance (LSD). !FTL, whose efficiency is demonstrated on 2D stroke gestures so far, should be equally applicable to 3D gestures.

The aforementioned algorithms benefit from a geometrical interpretation, which allows the recognition process to be explained while attaining relatively high recognition rates and using few reference gestures [17]. To explore these advantages, we propose a similar approach to the aforementioned recognizers inspired by stroke gesture recognition.

III. Materials

A. Wearable Electronics

The system used for acquiring data from gestures was based on a sensory glove and inertial measuring units (IMUs). In particular, the Hiteg sensory glove (developed by our Health Involved Technical Engineering Group) consists of a custom-made glove equipped with ten flex sensors (inserted into slim pockets) to capture the flexion angles of the metacarpophalangeal and proximal interphalangeal joints, with a measurement error of less than 4.7°[12], [38]. Through the inbuilt electronic circuitry, all signals are sampled with a 12-bit resolution, at 37.5 Hz, which is considered adequate for SLR [13], [24]. During data acquisition, the glove was worn on the dominant hand because, in many cases, the nondominant only aids to emphasize the meaning conveyed by it. Besides, two of the selected gestures are performed only with the dominant hand.

The motion capture and analysis system, termed Movit G1 (by Captiks Srl, Italy [39]) is composed of six IMUs and a wireless USB receiver. Each IMU is equipped with a 3-axis accelerometer and 3-axis gyroscope to capture acceleration and angular velocity at a rate of 40 Hz, with a proved measurement error of 5% when compared to other systems regarded as the "gold standard" [12]. The IMUs were worn on both hands, forearms, and arms by means of Velcro® straps.

B. Subjects

The LIS words were performed by five inexperienced non-native signers, three males and two females, aged 34.80±11.73 (SD) years old. All subjects were taught how to sign each of the words by watching videos of experienced LIS signers.

C. LIS Gestures

The chosen sign words were: "Ciao", "Cose", "Google", "Grazie", "Internet", "Jogging", "Maestra", "Pizza", "TV" and "Twitter", selected for their common use and, in some cases, for their international meaning. Although the selected vocabulary is limited to 10 words, we consider that it can provide a good generalization of LIS since the selected gestures are dynamic and involve different hand, arm and forearm movements. For instance, some gestures are signed with only one hand and arm (e.g.: "Twitter"), others with both hands and arms performing the same "mirrored" movement (e.g.: "TV"), and others with different movements for each limb (e.g., "Maestra").

IV. Methods

A. A Geometric Model-based Approach

1) Window Filtering and Data Scaling: Accelerometers, gyroscopes, and flex sensors capture data in distinct units of measure. However, the proposed algorithm relies on the sum of different types of point-wise distances to compare a candidate with a template gesture, which means that all data must be scaled to obtain normalized dimensionless metrics. Regarding the sensory glove, the digital values acquired with each of the flex sensors were observed to be approximately within a 500-4500 range of digital values (representing Ohmic resistance). Assuming that the flex sensors can be grouped per finger, the set $\mathcal{G}$ of sensor pairs is represented as:

$$\mathcal{G} = \{Thumb, Index, Middle, Ring, Little\} \tag{1}$$

Thus, a raw point $RawP_{(i)}^{(f)}$, where $f \in \mathcal{G}$, will lie within the rectangle $[500, 4500] \times [500, 4500] \subset \mathbb{R}^2$. Based on these limits, each selected point $RawP_{(i)}^{(f)}$ is centred and scaled as in (2), resulting in a scaled raw point $sRawP_{(i)}^{(f)}$ comprised in the square $[-1, 1] \times [-1, 1]$.

$$sRawP_{(i)}^{(f)} = \frac{RawP_{(i)}^{(f)} - \left(\frac{4500+500}{2},\ \frac{4500+500}{2}\right)}{\frac{4500-500}{2}} \in\ [-1,1] \times\ [-1,1]\ \subset \mathbb{R}^2 \tag{2}$$

Outliers in the IMU sensor data (which can occasionally occur due to possible gaps in the wireless connection) can result in misleading scaled values, thus they were filtered. To this aim, the 99.5$^{\text{th}}$ and 0.5$^{\text{th}}$ quantiles were calculated separately for each of the three axes, for all acquired data. Consequently, the range of accelerometer data was contained within the Cartesian window $[-1.21, 1.21] \times [-1.32, 1.28] \times [-0.80, 1.61] \subset \mathbb{R}^3$. Analogously, the considered gyroscope data lied within the Cartesian window $[-341.95, 345.18] \times [-304.33, 338.60] \times [-258.54, 261.83] \subset \mathbb{R}^3$. The set $\mathcal{IMU}$ of IMU sensors is represented in (3).

$$\mathcal{IMU} = \{acc_1, \dots, acc_6, gyro_1, \dots, gyro_6\} \tag{3}$$

To obtain dimensionless metrics, each raw point $RawP_{(i)}^{(acc_n)} = (x_i, y_i, z_i)$, of an accelerometer acc_n, is scaled and centered as in (4), considering the aforementioned limits of the respective Cartesian window, resulting in a scaled raw point $sRawP_{(i)}^{(acc_n)}$ comprised in the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$.

$$sRawP_{(i)}^{(acc_n)} = \left(\frac{x_i - \frac{1.21-1.21}{2}}{\frac{1.21+1.21}{2}}, \frac{y_i + \frac{1.28-1.32}{2}}{\frac{1.28+1.32}{2}}, \frac{z_i - \frac{1.61-0.80}{2}}{\frac{1.61+0.80}{2}}\right) \in [-1,1] \times [-1,1] \times [-1,1] \subset \mathbb{R}^3, n \in [1,2,...,6] \tag{4}$$

Similarly, each raw point $RawP_{(i)}^{(gyro_n)}c = (x_i, y_i, z_i)$, of a gyroscope $gyro_n$, was scaled and centered as in (5), considering the aforementioned limits of the respective Cartesian window, resulting in a scaled raw point $sRawP_{(i)}^{(gyro_n)}$ comprised in the cube $[-1, 1] \times [-1, 1] \times [-1, 1]$.

$$sRawP_{(i)}^{(gyro_n)} = \left(\frac{x_i - \frac{345.18-341.95}{2}}{\frac{345.18+341.95}{2}}, \frac{y_i + \frac{338.60-304.33}{2}}{\frac{338.60+304.33}{2}}, \frac{z_i - \frac{261.83-258.54}{2}}{\frac{261.83+258.54}{2}}\right) \in [-1,1] \times [-1,1] \times [-1,1] \subset \mathbb{R}^3, n \in [1,2,...,6] \tag{5}$$

2) Uniformly linearly interpolated resampling for d-dimensional trajectories: The list of points that represent each gesture may vary in length, as different users can perform the same gesture with different repeatability. To efficiently compare the two gestures, we considered the same number of points along both respective trajectories and applied a uniform linear interpolation method to resample the n scaled raw points of a trajectory to N equally spaced points, considering the distance along the trajectory instead of the Euclidean distance. By doing so, we aim at sampling the points that correspond to the same time instants of each gesture, regardless of the speed at which they were performed, thus allowing a point-wise comparison. Besides, the resampling allows independence of the algorithm to the input size. Considering a list of n scaled raw points coming from a certain d-dimensional sensor

$$sRawP = \left\{sRawP_{(1)}, \dots, sRawP_{(n)}\right\} \in \mathbb{R}^{nd} \quad (6)$$

We assumed that $n \geq 2$, and that at least two points in the list are distinct. To linearly interpolate N points from (6), we define the $n-1$ vectors

$$\vec{rv}_j = sRawP_{(j+1)} - sRawP_{(j)} \in \mathbb{R}^d, \forall j \in \{1, ..., n-1\} \quad (7)$$

connecting two consecutive scaled raw points of $sRawP$, and

$$\lambda_1 = 0 \quad (8)$$

$$\lambda_{k+1} = \lambda_k + |\vec{rv}_k|, \forall\ k \in \{1, \dots, n-1\} \quad (9)$$

where $|\vec{rv}_k|$ is the d-dimensional Euclidean distance between two consecutive points of $sRawP$, that is the Euclidean length of vector $\vec{rv}_k$. Thus,

$$\lambda = \lambda_n = \sum_{j=1}^{n-1} |\vec{rv}_j| \quad (10)$$

is the total length of the polygonal line joining the consecutive points of $sRawP$. We looked for a list sP of N points $sP_{(i)}$ uniformly lying on the polygonal line determined by $sRawP$. These points $sP_{(i)}$ can be obtained with expression (11), which is valid in every Euclidean Affine space regardless of its dimension d.

$$sP_{(h+1)} = sRawP_{(1)} + \sum_{k=1}^{n-1} min\left\{\left(h\frac{\lambda}{N-1} - \lambda_k\right)^+, |\vec{rv}_k|\right\} \frac{\vec{rv}_k}{||\vec{rv}_k||}, \text{ with } h \in \{0, \dots, N-1\} \quad (11)$$

where $\tau^+ = max\{\tau, 0\}$, is the positive part of the real number τ, and

$$||\vec{rv}_k|| = \begin{cases} 1, \text{if } |\vec{rv}_k| = 0 \\ |\vec{rv}_k|, \text{if } |\vec{rv}_k| \neq 0 \end{cases} \quad (12)$$

3) The multivector C^1 metric: Let sP (13) and sQ (14) be two lists of N resampled, centred, and scaled points in $\mathbb{R}^d$

$$sP = \left\{sP_{(1)}, \dots, sP_{(N)}\right\} \in \mathbb{R}^{Nd} \quad (13)$$

$$sQ = \left\{sQ_{(1)}, \dots, sQ_{(N)}\right\} \in \mathbb{R}^{Nd} \quad (14)$$

where d=2 for the glove data, and d=3 for the IMU data, obtained from a particular sensor when measuring two gestures $\mathcal{P}$ and $\mathcal{Q}$, respectively.

We define the point-wise distance $pwD_{(sP,sQ)}$, the point-wise vector distance $pwVD_{(sP,sQ)}$, and the point-wise multivector distance $pw\mu VD_{(sP,sQ)}$ between sP and sQ as:

$$pwD_{(sP,sQ)} = \frac{1}{N}\sum_{i=1}^{N}\left|sP_{(i)} - sQ_{(i)}\right| \quad (15)$$

$$pwVD_{(sP,sQ)} = \frac{1}{N-1}\sum_{i=1}^{N-1}\left|\vec{sv}_{(i)} - \vec{sw}_{(i)}\right| \quad (16)$$

$$pw\mu VD_{(sP,sQ)} = \frac{1}{(N-2)}\sum_{i=1}^{N-2}\left|\frac{\vec{sv}_{(i)} \cdot \vec{sv}_{(i+1)}}{\left|\vec{sv}_{(i)}\right|\left|\vec{sv}_{(i+1)}\right|} - \frac{\vec{sw}_{(i)} \cdot \vec{sw}_{(i+1)}}{\left|\vec{sw}_{(i)}\right|\left|\vec{sw}_{(i+1)}\right|}\right| + \frac{1}{(N-2)}\sum_{i=1}^{N-2}\left|\frac{\vec{sv}_{(i)} \wedge \vec{sv}_{(i+1)}}{\left|\vec{sv}_{(i)}\right|\left|\vec{sv}_{(i+1)}\right|} - \frac{\vec{sw}_{(i)} \wedge \vec{sw}_{(i+1)}}{\left|\vec{sw}_{(i)}\right|\left|\vec{sw}_{(i+1)}\right|}\right| \quad (17)$$

where

$$\vec{sv}_{(i)} = sP_{(i+1)} - sP_{(i)} \in \mathbb{R}^d, \forall\ i \in \{1, \dots, N-1\} \quad (18)$$

$$\vec{sw}_{(i)} = sQ_{(i+1)} - sQ_{(i)} \in \mathbb{R}^d, \forall\ i \in \{1, \dots, N-1\} \quad (19)$$

and

- $(x, y) \cdot (t, u) = xt + yu \in \mathbb{R}$ if $d = 2$;
- $(x, y, z) \cdot (t, u, v) = xt + yu + zv \in \mathbb{R}$ if $d = 3$;
- $(x, y) \wedge (t, u) = xu - yt \in \mathbb{R}$ if $d = 2$,
- $(x, y, z) \wedge (t, u, v) = (yv - zu, zt - xv, xu - yt) \in \mathbb{R}^3$ if d=3,

where $x, y, z, t, u, v \in \mathbb{R}$.

We observe that pwD is the sum of Euclidean distances between points in $\mathbb{R}^d$, and $pwVD$ is the sum of Euclidean distances between vectors in $\mathbb{R}^d$. The elaborated quantities

$$PS_{(i)} = \left(\frac{\vec{sv}_{(i)} \cdot \vec{sv}_{(i+1)}}{\left|\vec{sv}_{(i)}\right|\left|\vec{sv}_{(i+1)}\right|}, \frac{\vec{sv}_{(i)} \wedge \vec{sv}_{(i+1)}}{\left|\vec{sv}_{(i)}\right|\left|\vec{sv}_{(i+1)}\right|}\right) \in \mathbb{R}^{\frac{d(d-1)}{2}+1} \quad (20)$$

$$QS_{(i)} = \left(\frac{\vec{sw}_{(i)} \cdot \vec{sw}_{(i+1)}}{\left|\vec{sw}_{(i)}\right|\left|\vec{sw}_{(i+1)}\right|}, \frac{\vec{sw}_{(i)} \wedge \vec{sw}_{(i+1)}}{\left|\vec{sw}_{(i)}\right|\left|\vec{sw}_{(i+1)}\right|}\right) \in \mathbb{R}^{\frac{d(d-1)}{2}+1} \quad (21)$$

encode the normalized shapes [18] of the oriented triangle determined by two consecutive vectors ($\vec{sv}_{(i)}$, $\vec{sv}_{(i+1)}$ and $\vec{sw}_{(i)}$, $\vec{sw}_{(i+1)}$, respectively) along the resampled trajectory. $PS_{(i)}$ and $QS_{(i)}$ can also be thought of as the encoding of d-dimensional rotation determined by two consecutive normalized vectors $\left(\frac{\vec{sv}_{(i)}}{|\vec{sv}_{(i)}|}, \frac{\vec{sv}_{(i+1)}}{|\vec{sv}_{(i+1)}|}\right.$ and $\frac{\vec{sw}_{(i)}}{|\vec{sw}_{(i)}|}, \frac{\vec{sw}_{(i+1)}}{|\vec{sw}_{(i+1)}|}$, $\left.\text{respectively}\right)$ along the resampled trajectory. For every couple of sensors per finger and for all accelerometers and gyroscopes, we propose the dimensionless metric μC^1 (22).

$$\mu C^1_{(sP,sQ)} = pwD_{(sP,sQ)} + pw\mu VD_{(sP,sQ)} \quad (22)$$

Before the experiments, it seemed reasonable to consider $pwVD$ as a term for μC^1. Surprisingly, the experimental results showed that $pwVD$ gives no significant contribution to recognition. This evidence will be discussed later in section V "Results and Discussion".

4) The multivector C^2 metric: A second multivector metric based on the concept of gesture "shape" [40], [41] was explored. As in Section C, we can consider sP, sQ, $\vec{sv}_{(i)}$ and $\vec{sw}_{(i)}$ to define $PS_{(i)}$ (20) and $QS_{(i)}$ (21) as the normalized shapes of sP and sQ, respectively.

We observe that $PS_{(i)}$, $QS_{(i)} \in \mathcal{S}^1 \subset \mathbb{R}^2$ for the case of the glove, whereas $PS_{(i)}$, $QS_{(i)} \in \mathcal{S}^3 \subset \mathbb{R}^4$ for the IMU data, where $\mathcal{S}^k = \{z \in \mathbb{R}^{k+1} : |z| = 1\}$. So, we can define the point-wise multivector Euclidean distance $pw\mu ED_{(sP,sQ)}$ between two normalized gesture shapes $PS_{(i)}$ and $QS_{(i)}$.

$$pw\mu ED_{(sP,sQ)} = \frac{1}{(N-2)} \sum_{i=1}^{N-2} \left| PS_{(i)} \ - \ QS_{(i)} \right| \quad (23)$$

which is just a slight modification of $pw\mu VD$. Nevertheless, this suggests to consider the shapes $PS_{(i)}$ as points on a trajectory having shapes in its turn (shapes of shapes, indeed). Let us go further with this idea just on the IMU data. We can define the normalized "shapes of shapes" as follows:

$$PSS_{(i)} = \left(\frac{PS_{(i)} \cdot s\vec{v}v_{(i)}}{\left| s\vec{v}v_{(i)} \right|}, \frac{PS_{(i)} \wedge s\vec{v}v_{(i)}}{\left| s\vec{v}v_{(i)} \right|} \right) \in \mathcal{S}^6 - (1,0,0,0,0,0,0) \ \subset \mathbb{R}^7 \quad (24)$$

$$QSS_{(i)} = \left(\frac{QS_{(i)} \cdot s\vec{w}w_{(i)}}{\left| s\vec{w}w_{(i)} \right|}, \frac{QS_{(i)} \wedge s\vec{w}w_{(i)}}{\left| s\vec{w}w_{(i)} \right|} \right) \in \mathcal{S}^6 - (1,0,0,0,0,0,0) \ \subset \mathbb{R}^7 \quad (25)$$

with $i = 1, \ldots, N-3$, and where $s\vec{v}v_{(i)}$ and $s\vec{w}w_{(i)}$ are defined as follows:

$$s\vec{v}v_{(i)} = PS_{(i+1)} - PS_{(i)} \ \in \mathbb{R}^4, \forall\, i \in \{1, \ldots, N-3\} \quad (26)$$

$$s\vec{w}w_{(i)} = QS_{(i+1)} - QS_{(i)} \ \in \mathbb{R}^4, \forall\, i \in \{1, \ldots, N-3\} \quad (27)$$

where

- $(a,x,y,z) \cdot (b,t,u,v) = ab + xt + yu + zv \in \mathbb{R}$;
- $(a,x,y,z) \wedge (b,t,u,v) = (at-bx, by-au, av-bz, xu-yt, zt-xv, yv-zu) \in \mathbb{R}^6$, $\forall\ a,x,y,z,b,t,u,v \in \mathbb{R}$,
- $|\cdot|$ is the usual Euclidean norm in $\mathbb{R}^4$,

$$|(\alpha_1, \cdots, \alpha_4)| = \sqrt{(\alpha_1)^2 + \cdots + (\alpha_4)^2} \ .$$

Now, the normalized shape $PSS_{(i)}$ is computed with respect to the oriented triangle made up by vector $PS_{(i)}$ (on the circle of normalized shapes) and vector $s\vec{v}v_{(i)}$ (the vector joining two consecutive shapes). This forces the values of the scalar product $\frac{PS_{(i)} \cdot s\vec{v}v_{(i)}}{\left| s\vec{v}v_{(i)} \right|}$ to lie between -2 and 0, explaining why such "shapes of shapes" lie on the translated sphere $\mathcal{S}^6 - (1,0,0,0,0,0,0)$. Now, we introduce the point-wise second-order multivector Euclidean distance $pw\mu^2 ED_{(sP,sQ)}$ as follows:

$$pw\mu^2 ED_{(sP,sQ)} = \frac{1}{(N-3)} \sum_{i=1}^{N-3} \left| PSS_{(i)} \ - \ QSS_{(i)} \right| \quad (28)$$

It is worth pointing out that a similar $pw\mu^2 ED_{(sP,sQ)}$ can also be defined for the glove. Accordingly, we tried to do this, but, differently from the IMUs, a negative effect on the recognition rate was observed, so it was excluded.

Finally, we defined the dimensionless metric μC^2 as

$$\mu C^2_{(sP,sQ)} = pwD_{(sP,sQ)} + pw\mu ED_{(sP,sQ)} + pw\mu^2 ED_{(sP,sQ)} \quad (29)$$

where $pwD_{(sP,sQ)}$ is defined in the previous section.

5) Global Distances: With a certain pair of gestures $\mathcal{P}$ and $\mathcal{Q}$, corresponds the two lists of N resampled and scaled points $sP^{(f)}$ and $sQ^{(f)}$, organized in the sets $\{sP^{(f)}\}_{f \,\in\, \mathcal{G} \cup \mathcal{IMU}}$ and $\{sQ^{(f)}\}_{f \,\in\, \mathcal{G} \cup \mathcal{IMU}}$, respectively.

Thus, regarding the metric μC^1, the dissimilarity between both gestures can be measured with the global metric (30).

$$G\mu C^1{}_{(\mathcal{P},\mathcal{Q})} = \sum_{f \in \mathcal{G} \cup \mathcal{IMU}} \mu C^1{}_{(sP^{(f)}, sQ^{(f)})} \quad (30)$$

Analogously, the dissimilarity between gestures can be also measured with the global μC^2 metric (31).

$$G\mu C^2{}_{(\mathcal{P},\mathcal{Q})} = \sum_{f \in \mathcal{G} \cup \mathcal{IMU}} \mu C^2{}_{(sP^{(f)}, sQ^{(f)})} \quad (31)$$

Thus, to classify a given candidate gesture, $G\mu C^1$ (or $G\mu C^2$) was calculated between the latter and each template gesture in the training set. Considering both $G\mu C^1$ and $G\mu C^2$ measure dissimilarity, the candidate is given by the class of the template gesture that minimizes $G\mu C^1$ (or $G\mu C^2$).

B. Data Acquisition

Each of the 10 LIS words was signed 100 times by each subject , a relatively high number for a SLR dataset. Moreover, all repetitions started and ended with the hands resting on the tights. Thus, the acquired dataset consisted of 5000 sign words, of which 30% was considered as the training set and the remaining 70% as test set. Data were acquired with the aid of Matlab software (by MathWorks, Inc.) and Captiks Motion Studio® (by Captiks, Rome, Italy).

C. Experimental Design

The μC^1 and μC^2 algorithms were evaluated on the LIS dataset in both user-dependent and user-independent scenarios. Moreover, the algorithms' performances were compared with other two benchmark classifiers, SVM and k-NN, already used for SLR [42], along with two SotA DL models, BiLSTM and GRU. Evaluation was performed through Matlab on a PC equipped with a Nvidia® GeForce® MX150 graphics card, an Intel® Core™ i7-8565U processor and 16 Gb of RAM.

1) Parameter Optimization: According to the description of μC^1 and μC^2 in Section IV-A, we considered two parameters that can be tuned to optimize the algorithm's performance: the N resampled points and the T training examples (templates) per class. For their optimization, we performed an adapted cross-validation procedure on the train set data. In each iteration of this adapted cross-validation, the model was trained with T training examples of each gesture, randomly selected from the data belonging to four of the subjects. This model was then validated on the data from the remaining subject. The process was repeated until each of the subjects was used for validation. Because the total $10 \times T$ training examples were randomly selected, the adapted cross-validation was repeated 10 times for statistical purposes. Data used for training and validation always belongs to separate subjects, thus ensuring user-independent conditions.

The adapted cross-validation was combined with a grid search technique. The values considered for N ranged from

6 to 20, equally spaced by 2, whereas the values for T ranged from 1 to 22, equally spaced by 3. Both ranges were determined empirically by direct observation of the recognition rates and recognition times for different values of N and T. The choice of the optimal parameters was based on the mean recognition rate and mean recognition time (i.e., the amount of time needed to pre-process and classify a candidate gesture).

2) User-Dependent Evaluation: User-dependent evaluation is achieved by performing a 7-fold inverse cross-validation on each subject, considering only the data in the test set. Since in each iteration the larger fold is used for validation and the smaller for training, an inverse cross-validation [43] can be taken as an adequate method to evaluate the performance of a model trained with a relatively low amount of data. The number of folds was chosen according to the optimal T and the number of candidate gestures per subject in the test set (approximately 700).

The performance of each algorithm was assessed in terms of recognition rate, along with training time (time to train the model), pre-processing time (time to preprocess a candidate gesture, i.e., scaling and interpolating), classification time (time to compare a gesture candidate against each training template) and recognition time (sum of the pre-processing and classification times).

3) User-Independent Evaluation: The previously mentioned adapted cross-validation approach for parameter optimization was also found to be adequate for user-independent evaluation. However, in each iteration, the model was validated on the remaining subject's data taken from the test set, instead of the data taken from the train set. This was done with the aim of eliminating any bias, since the parameters were optimized with train set data. Moreover, instead of 10 times, the adapted cross-validation was repeated 100 times for each algorithm. The performance of each algorithm was assessed using the same metrics as in the user-dependent scenario.

V. Results and Discussion

A. Models and parameter tuning

As N and T increase, the recognition rate rapidly reaches a plateau near the 90% mark for both algorithms (Fig. 1). In contrast, recognition time continues to have a steady increase throughout both parameter ranges, although this is more noticeable for μC^2. A small increment in the recognition rate is also observed for higher N and T values, however the cost in terms of recognition time is disproportionate.

Keeping this in mind, N=12 and T=10 were considered to be the optimal pair of parameters for the usage of both μC^1 μC^2 on the given dataset, for which a mean recognition rate of 89.88% and 91.59% and a mean recognition time of 7.55 ms and 11.43 ms were registered, respectively.

To establish a fair basis of comparison between all classifiers, the values of these parameters were kept the same when using the remaining classifiers, considering the N resampled points were reorganized and handled as 1-dimensional feature vectors for the k-NN and SVM. The sequences of resampled points were organized as 2D matrices of size 12×46 (12×3 sequences for each accelerometer/gyroscope and 12×2 sequence for each finger) for BiLSTM and GRU input.

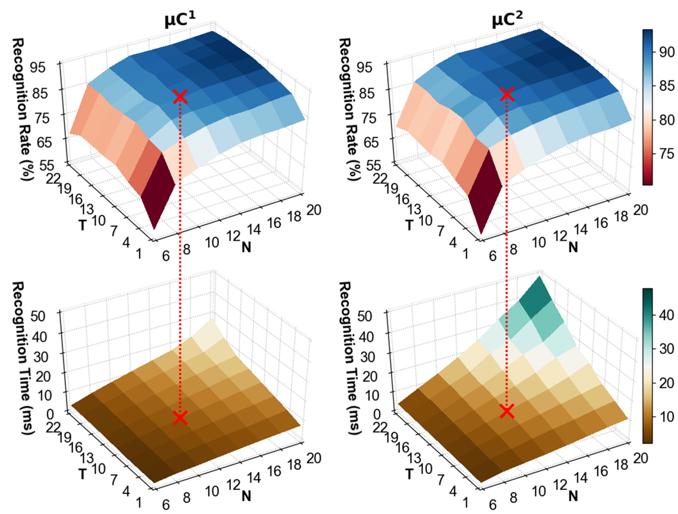

Fig. 1. Surface plot of the mean recognition rate and recognition time per pair of parameters T and N, regarding μC^1 (left) and μC^2 (right).

Both SVM and k-NN have parameters that require tuning. The cost parameter C for the linear SVM and the number of k nearest neighbors for the k-NN. Using the same adapted cross-validation approach as before, C varied from 10^{-4} to 10^4 (logarithmically spaced), whereas k varied from 1 to 19 with intervals of 2. Since both parameters have a negligible effect on the recognition time, this metric was not taken into account for selecting the optimal values. The optimal values of C and k were found to be 0.1 and 1, respectively.

The implemented BiLSTM and GRU architectures were based on the work by Zhao et al. [44], where both model architectures are illustrated and were used to classify IMU-based data resulting in high recognition rates. The BiLSTM was composed of two hidden BiLSTM layers, both with 64 hidden units, whereas the GRU was composed of two hidden GRU layers, the first with 128 hidden units and the second with 64. Additionally, each of the networks' output was followed by a fully connected layer with 10 hidden units (corresponding to the number of classes) and a SoftMax activation function. For both networks, we used the Adam optimizer and the number of training epochs were set to 200, using a mini batch size of 32. As in [44], the learning rate was set to 0.001.

To avoid overfitting while training the DL models, for user-independent evaluation we used a validation set composed of 10 samples of each gesture taken from the training set, for which the loss was evaluated every 50 training iterations. Training was stopped if the loss did not improve after 5 consecutive evaluations.

B. Recognition Rate

For all variables, we computed four normality tests: K-sample Anderson-Darling test, D'Agostino K^2 test, Shapiro-Wilk W test, and Kolmogorov-Smirnov *KS*. Recognition rates of the four recognizers do not follow a normal distribution since they all fail at least one test for normality out of the four computed. Thus, we compute a Friedman test with

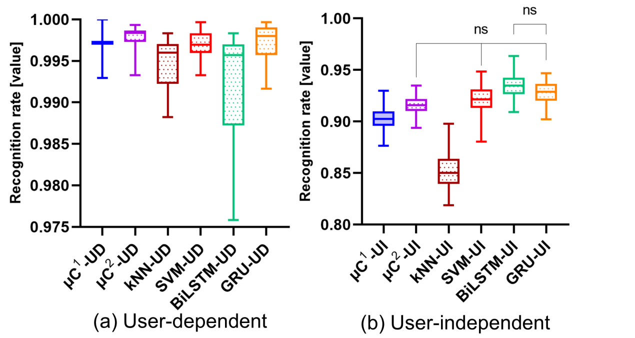

Fig. 2. Recognition rates in: (a) user-dependent, (b) -independent scenarios.

Dunn's multiple comparison on repeated measures since cross-validations are identical for each test.

In the user-dependent scenario (Fig. 2a), μC^2 (*M*=99.77%, *SD*=0.2025%) outperforms all competitors: μC^1 (*M*=99.70%, *SD*=0.2072%), SVM (*M*=99.69%, *SD*=0.2072%), k-NN (*M* = 99.46%, *SD*=0.3544%), BiLSTM (*M*=99.22%, *SD*=0.8059%) and GRU (*M*=99.70%, *SD*=0.2797%). μC^2 benefits from the smallest standard deviation (SD), thus suggesting that it remains constant in delivering its average recognition rate. No recognizer statistically differs from others (*F*=9.041, *p*=.1074, *n.s.*). In conclusion, μC^2 is the most accurate in user-dependent scenario, but not significantly more than others.

In the user-independent scenario (Fig. 2b), BiLSTM (*M*= 93.43%, *SD*=0.1026%) is the most accurate recognizer, followed by GRU (*M*=92.79%, *SD*=0.1092%), SVM (*M*=92.09%, *SD*=0.1282%), and μC^2 (*M*=91.56%, *SD*=0.0929%). Again, μC^2 benefits from the smallest SD, thus suggesting that it does not deviate much from its average recognition rate. SVM, although superior by only .53% to μC^2, is not significantly better than μC^2 (*Z*=1.550, *n.s.*). In addition, μC^2 is very highly significantly more accurate than μC^1 (*Z*=5.405, $p<.0001^{****}$) and k-NN (*Z*=10.02, $p<.0001^{****}$). On the other hand, BiLSTM (*Z*=6.274, $p<.0001^{****}$) and GRU (*Z*=4.195, $p=.0004^{***}$) are both significantly more accurate than μC^2, although for only 1.87% and 1.23%, respectively. Moreover, there are no significant differences between BiLSTM and GRU (*Z*=2.079, $n.s.$) and GRU and SVM (*Z*=2.646, $n.s.$).

C. Training Time

Training times do not follow a normal distribution, thus a Friedman test was again computed with repeated measures.

In the user-dependent scenario (Fig. 3), k-NN (*M*=0.1341 sec, *SD*=0.002386 sec) is slightly faster to train than μC^2 (*M*=0.1356 sec, *SD*=0.007278 sec), but not significantly (*Z*=0.2070, *n.s.*). Then come μC^1 (*M*=0.1456 sec, *SD*=0.008963 sec) and SVM (*M*=0.4523 sec, *SD*=0.1137 sec) respectively. SVM is significantly slower than k-NN (*Z*=3.726, $p=.001^{**}$) and μC^2 (*Z*=3.519, $p=.003^{**}$). No other significant differences were detected.

In the user-independent scenario (Fig. 3), μC^2 (*M*=0.1314 sec, *SD*=0.003970 sec) is faster to train than k-NN (*M*=0.1325 sec, *SD*=0.01896 sec), but not significantly (*Z*=0.8216, *n.s.*). SVM (*M*=0.3793 sec, *SD*=0.03209 sec) is the slowest recognizer to train among all candidates, and very highly significantly ($p<.0001^{****}$). μC^2 is also significantly faster than its counterpart μC^1 (Z=4.108, p=.0002***). Other significant differences are depicted in Fig. 3.

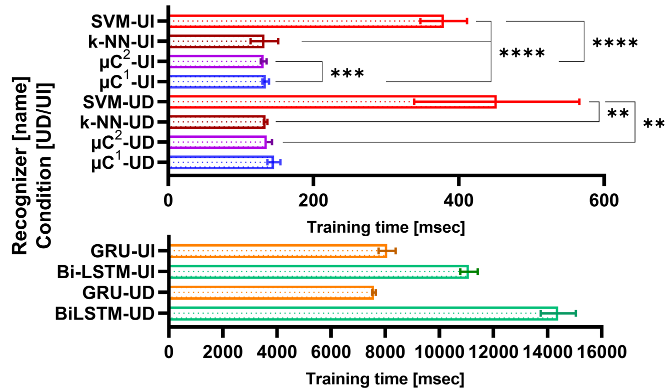

Fig. 3. Training time in user-dependent (UD), -independent (UI) scenarios.

Due to their much higher duration, the training times regarding the BiLSTM and GRU are depicted in a separate plot for both evaluation scenarios.

D. Pre-processing Time

Since pre-processing times also do not follow a normal distribution for both evaluation scenarios, a Friedman test was again computed. In the user-dependent scenario (Fig. 4), the Friedman test revealed a very highly significant difference between the six conditions (*F*=34.02, $p<.0001^{****}$). More precisely, k-NN (*M*=1.208 msec, *SD*=0.01821 msec) and GRU (*M*=1.208 msec, *SD*=0.00851 msec) are the fastest to pre-process a candidate gesture , followed by SVM (*M*=1.276 msec, *SD*=0.02248 msec), μC^1 (*M*=1.640 msec, *SD*=0.03392 msec), μC^2 (*M*=1.780 msec, *SD*=0.01061 msec) and BiLSTM (*M*=1.994 msec, *SD*=0.19430 msec). BiLSTM is not significantly faster than μC^1 (*Z*=2.0000, *n.s.*) and μC^2, (*Z*=1.0000, *n.s.*) whereas GRU is significantly faster than μC^2 (*Z*=3.5710, p=0.0053**).

In the user-independent scenario, there are significant differences between all classifiers, except between the SVM and both BiLSTM (*Z*=2.7970, *n.s.*) and GRU (*Z*=2.117, *n.s.*).

The higher pre-processing time regarding μC^1 and μC^2 can be justified by the need of organizing the candidate gesture data in the input format required by both algorithms, which differs from the other classifiers. Differently, the higher BiLSTM pre-processing time in user-dependent evaluation might have occurred due to parallel Windows processes which may have compromised CPU performance.

E. Classification Time

Again, since classification times did not pass normality tests, we computed the Friedman test with Dunn' multiple comparison on repeated measures as usual.

In the user-dependent scenario (Fig. 5), k-NN (*M*=0.7207 msec, *SD*=0.01436 msec) turns out to be the fastest recognizer to classify a candidate, followed by GRU (*M*=2.640 msec, *SD*=0.01741 msec), BiLSTM (*M*=4.417 msec, *SD*=0.40820

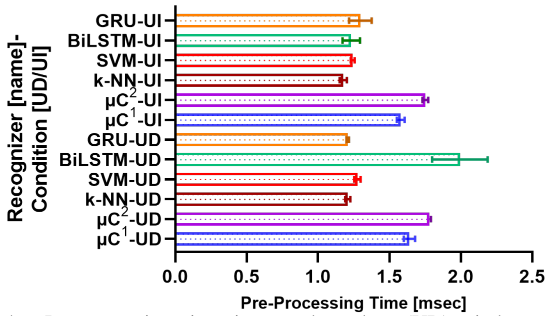

Fig. 4. Pre-processing time in user-dependent (UD), -independent (UI) scenarios.

msec), μC^1 (*M*=6.031 msec, *SD*=0.02097 msec), μC^2 (*M*=9.786 msec, *SD*=0.03652 msec), and SVM (*M*=12.32 msec, *SD*=0.2290 msec). However, k-NN is not significantly faster than μC^1 (p=.8838, *n.s.*) and μC^2 is not significantly faster than SVM (p=.8838, *n.s.*). μC^1 is not significantly slower than BiLSTM (p>.9999, *n.s.*) and GRU (p=0.6825., *n.s.*). GRU is significantly faster than μC^2 (p=.0405*). BiLSTM classification time is higher probably due to the reasons stated in Section V-E. The order for the user-independent scenario remains the same as in the user-dependent scenario. In this case significant differences were found between all classifiers, except between BiLSTM and GRU (p>.9999, *n.s.*).

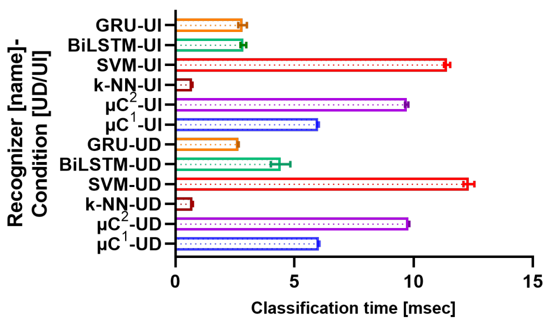

Fig. 5. Classification time in user-dependent (UD), -independent (UI) scenarios.

F. Overall Comparison

Fig. 6 sums up the individual results obtained for separate scenarios into an averaged measure (UD+UI=user-dependent and independent). The left part depicts the recognition rate, while the right part depicts the global recognition time, which is decomposed into pre-processing time and classification time.

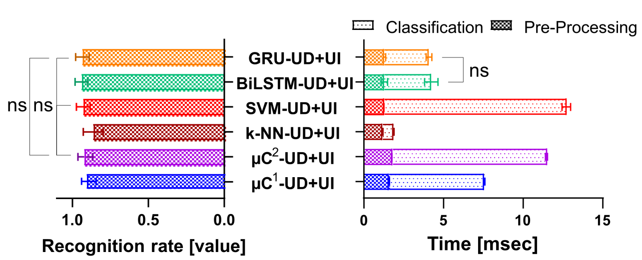

Fig. 6. Overall Comparison of the four recognizers: Recognition rate (left) vs. Recognition time (right). Error bars show a confidence interval of 95%.

Regarding the recognition rate, BiLSTM (*M*=93.81%, *SD*=4.170%), GRU (*M*=93.24%, *SD*=4.487%) and SVM (*M*=92.59%, *SD*=4.598%) slightly (by 1.71%, 1.14% and 0.49%, respectively) outperforms μC^2 (*M*=92.10%, *SD*=4.014%), but not significantly for the GRU (*Z*=2.0750, *n.s.*) and SVM (*Z*=0.1716, *n.s.*). In turn, μC^2 very highly significantly (p<.00001****) outperforms μC^1 (*M*=90.86%, *SD*=4.578%) and k-NN (*M*=86.07%, *SD*=6.565%). μC^2 also benefits from the smallest SD among the four recognizers.

Regarding the recognition time, k-NN (*M*=1.879 msec, *SD*=0.047 msec) is the fastest recognizer and very highly significantly faster than all other classifiers. In fact, all classifiers present significant differences between each other, with the exception of GRU and BiLSTM (*Z*=0.1827, *n.s.*).

Thus, μC^2 offers a good compromise between recognition rate and time. Regarding the recognition rate, it is even comparable with SotA DL methods such as GRU (with which no significant differences were found when considering UD+UI) and BiLSTM. Furthermore, since the recognition rate scored higher than 90% in both user-dependent and user-independent evaluation, it can be considered adequate for real-time interaction [45]). SVM is the slowest recognizer among the four conditions and k-NN is the fastest recognizer, however all classifiers remain under the threshold of 0.1 sec, the ideal response time for direct manipulation of user interfaces [46]. Besides, both μC^1 and μC^2 benefit of very low training times (which is only based in pre-processing all the gestures in the training set), much lower than BiLSTM and GRU, which can enhance the user experience in application than require fast calibration and allow the integration of new training examples on-the-fly. Furthermore, since no complex computations are necessary, these algorithms can be implemented in devices with lower computational power and limited memory space.

Fig. 7 depicts the average normalized confusion matrices of all the classifiers regarding user-independent evaluation. As is can be observed, some algorithms are able to recognize certain gestures more accurately than others. For instance, both μC^2 and μC^1 recognize "Google" and "Internet" better than BiLSTM and GRU, although these present better overall accuracy. Indeed, BiLSTM and GRU are advanced approaches that automatically compute and select features to optimize accuracy. However, both algorithms lack in interpretation and it is hard to fully grasp and visualize why the algorithm makes certain decisions and confuses some gestures with others. In contrast, μC^2 and μC^1 present the advantage to be based on geometric algebra, which can allow to understand how the recognition process takes place and how certain gestures can be confused with others. The approach is analytical, no longer statistical.

To demonstrate this, we used GeoGebra software to visualize situations where the recognition succeeded/failed. For instance, for μC^1 we present an example (Fig. 8a) regarding a couple of flex sensors from the index finger. Considering *Ciao1* as the candidate gesture, we can observe that $pwD_{Ciao1,Ciao2} > pwD_{Ciao1,Maestra1}$, which means that the candidate would be mistakenly recognized as "Maestra". Indeed, the points from *Ciao1* and *Maestra1* appear to be closer to each other. However, after summing $pw\mu VD$, we have $\mu C^1_{Ciao1,Ciao2} < \mu C^1_{Ciao1,Maestra1}$, allowing the candidate gesture to be correctly classified as "Ciao".

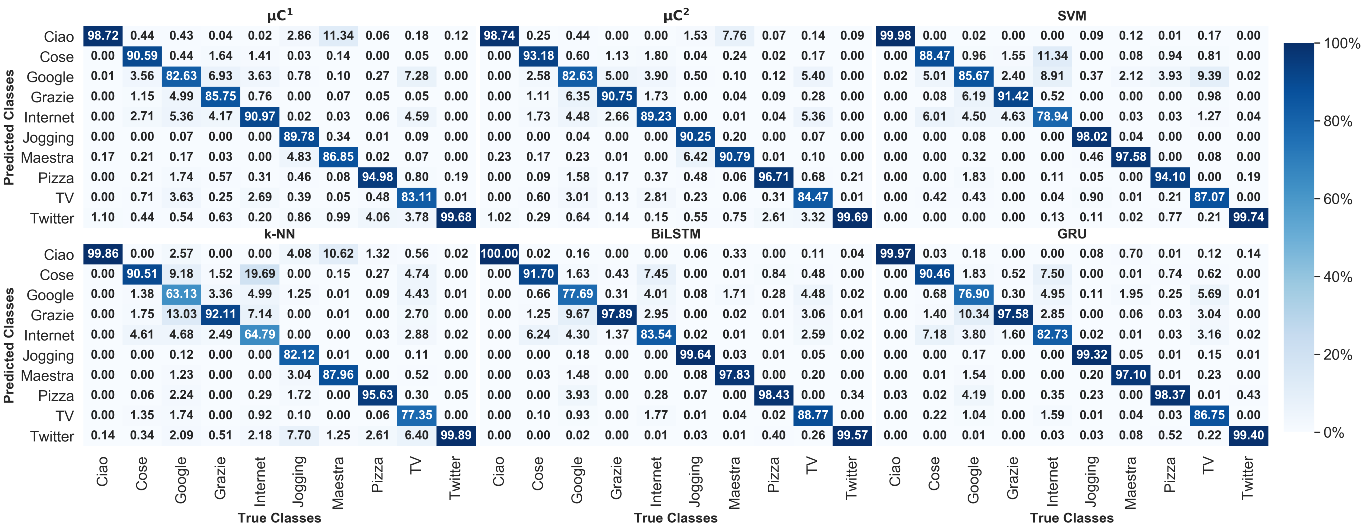

Fig. 7. Average normalized confusion matrices of all the classifiers regarding user-independent evaluation.

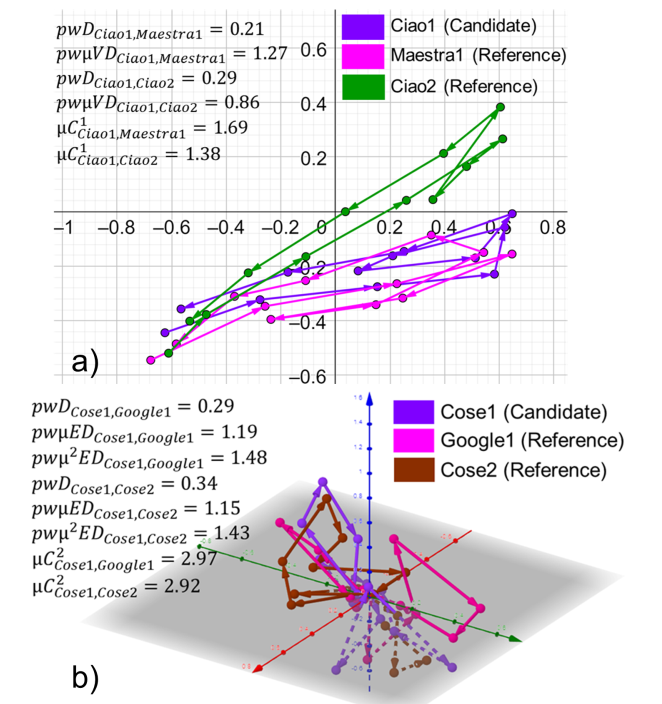

Fig. 8. a) μC^1 recognition example (index finger's couple of flex sensors): *Ciao1* is the candidate gesture, whereas *Maestra1* and *Ciao2* are two reference gestures contained in the training set; b) μC^2 recognition example (gyroscope worn on the right forearm): *Cose1* is the candidate gesture, whereas *Google1* and *Cose2* are two reference gestures contained in the training set.

Analogously, we present another example for μC^2 (Fig. 8b). This example depicts a similar situation regarding the right forearm's gyroscope. In this case pwD is also not sufficient to correctly classify the candidate gesture *Cose1*. Although $pw\mu ED_{Cose1,Cose2} < pw\mu ED_{Cose1,Google1}$, summing pwD and $pw\mu ED$ results in misclassification. However, by including $pw\mu^2 ED$, we have $\mu C^2_{Cose1,Cose2} < \mu C^2_{Cose1,Google1}$. Situations such as these help to justify why μC^2 recognizes most classes more accurately than μC^1 (Fig. 7).

The use of $pwVD$ in μC^1 seemed intuitive, however, when this metric was considered, the recognition rate of μC^1 surprisingly decreased to 87.60±3.31%. Due to this observation – and to reduce the computation time – $pwVD$ was excluded. Analogously, $pw\mu^2 ED$ was not considered for the flex sensor data due to its negative effective on the recognition rate, which dropped to 90.29±2.25%.

VI. Conclusion

We presented μC^1 and μC^2, two new gesture recognition geometric-based models inspired by stroke gesture recognition and Clifford algebra. Both algorithms were evaluated on a challenging custom sign language dataset, along with k-NN, SVM, BiLSTM and GRU for comparison purposes. Overall, considering both user-dependent and user-independent scenarios, μC^2 achieved a recognition rate with no significant differences from the ones obtained with the SVM and the GRU. Although the BiLSTM presented significantly higher accuracies, μC^2 offers the advantage of lower training times, simpler implementation and higher interpretability.

In sum, we presented a recognition algorithm that takes the advantages of NNC and performs with high recognition rates ($\geq$ 90% [45]) and low recognition times ($<$ 0.1 sec [46]), which satisfy the constraints for real-time gesture interaction. Following the current trend of XAI, the proposed algorithms offer the advantage of being explainable due to their relatively simple implementation based on geometric algebra. This was demonstrated by describing and visualizing the recognition process, allowing the user to understand the logic behind each decision. Regardless, the authors recognize the limitations of the used dataset, thus future work will include the evaluation of these algorithms on different datasets.

References

[1] G. Saggio, F. Giannini, M. Todisco, and G. Costantini, "A data glove based sensor interface to expressively control musical processes," in *Proc. of 4th IEEE Int. Workshop on Advances in Sensors and Interfaces (IWASI '11)*, 2011, pp. 192–195. DOI: 10.1109/IWASI.2011.6004715.

[2] G. Saggio, A. Lazzaro, L. Sbernini, F. M. Carrano, D. Passi, A. Corona, V. Panetta, A. L. Gaspari, and N. Di Lorenzo, "Objective surgical skill assessment: An initial experience by means of a sensory glove paving the way to open surgery simulation?" *Journal of Surgical Education*, vol. 72, no. 5, pp. 910–917, 2015. DOI: https://doi.org/10.1016/j.jsurg.2015.04.023.

[3] L. Sbernini, L. R. Quitadamo, F. Riillo, N. D. Lorenzo, A. L. Gaspari, and G. Saggio, "Sensory-glove-based open surgery skill evaluation," *IEEE Transactions on Human-Machine Systems*, vol. 48, no. 2, pp. 213–218, 2018. DOI: 10.1109/THMS.2017.2776603.

[4] V. Errico, M. Ricci, A. Pallotti, F. Giannini, and G. Saggio, "Ambient assisted living for tetraplegic people by means of an electronic system based on a novel sensory headwear: Increased possibilities for reduced abilities," in *Proc. of IEEE Int. Symposium on Medical Measurements and Applications (MeMeA '18)*, 2018, pp. 1–6. DOI: 10.1109/MeMeA.2018.8438638.

[5] G. Saggio, M. De Sanctis, E. Cianca, G. Latessa, F. De Santis, and F. Giannini, "Long term measurement of human joint movements for health care and rehabilitation purposes," in *Proc. of 1st Int. Conf. on Wireless Communication, Vehicular Technology, Information Theory and Aerospace Electronic Systems Technology*, 2009, pp. 674–678. DOI: 10.1109/WIRELESSVITAE.2009.5172528.

[6] G. Saggio, F. Cavrini, and C. A. Pinto, "Recognition of arm-and-hand visual signals by means of svm to increase aircraft security," in *Computational Intelligence*, J. J. Merelo, A. Rosa, J. M. Cadenas, A. D. Correia, K. Madani, A. Ruano, and J. Filipe, Eds., Cham: Springer Int. Publishing, 2017, pp. 444–461.

[7] R. M. Krauss, Y. Chen, and P. Chawla, "Nonverbal behavior and nonverbal communication: What do conversational hand gestures tell us?" In, ser. Advances in Experimental Social Psychology, M. P. Zanna, Ed., vol. 28, Academic Press, 1996, pp. 389–450. DOI: https://doi.org/10.1016/S0065-2601(08)60241-5.

[8] I. Dwijayanti, K. Budayasa, and T. Y. E. Siswono, "Objective surgical skill assessment: An initial experience by means of a sensory glove paving the way to open surgery simulation?" *Beta: Jurnal Tadris Matematika*, vol. 12, no. 2, pp. 133–143, 2019. DOI: https://doi.org/10.20414/betajtm.v12i2.307.

[9] P. De Silva, M. Osano, A. Marasinghe, and A. Madurapperuma, "Towards recognizing emotion with affective dimensions through body gestures," in *Proc. of 7th Int. Conf. on Automatic Face and Gesture Recognition (FGR '06)*, 2006, pp. 269–274. DOI: 10.1109/FGR.2006.106.

[10] W. Wundt, *The Language of Gestures*. De Gruyter Mouton, 2010. DOI: doi:10.1515/9783110808285.

[11] Q. Zhang, D. Wang, R. Zhao, and Y. Yu, "Myosign: Enabling end-to-end sign language recognition with wearables," in *Proc. of the 24th Int. Conf. on Intelligent User Interfaces*, ser. IUI '19, Marina del Ray, California: Association for Computing Machinery, 2019, pp. 650–660. DOI: 10.1145/3301275.3302296.

[12] G. Saggio, F. Tombolini, and A. Ruggiero, "Technology-based complex motor tasks assessment: a 6-DOF inertial-based system vs. a gold-standard optoelectronic-based one," *IEEE Sensors Journal*, vol. 21, no. 2, pp. 1616–1624, 2021, ISSN: 1530-437X. DOI: 10.1109/jsen.2020.3016642.

[13] G. Saggio, P. Cavallo, M. Ricci, V. Errico, J. Zea, and M. E. Benalcázar, "Sign Language Recognition Using Wearable Electronics: Implementing k-Nearest Neighbors with Dynamic Time Warping and Convolutional Neural Network Algorithms," *Sensors*, vol. 20, no. 14, p. 3879, 2020. DOI: 10.3390/s20143879.

[14] M. Asadi-Aghbolaghi, A. Clapes, M. Bellantonio, H. J. Escalante, V. Ponce-Lopez, X. Baro, I. Guyon, S. Kasaei, and S. Escalera, "A Survey on Deep Learning Based Approaches for Action and Gesture Recognition in Image Sequences," in *Proc. of the 12th IEEE Int. Conf. on Automatic Face and Gesture Recognition (FG '17) - 1st Int. Workshop on Adaptive Shot Learning for Gesture Understanding and Production (ASL4GUP '17)*, 2017, pp. 476–483. DOI: 10.1109/FG.2017.150.

[15] A. Barredo Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. Garcia, S. Gil-Lopez, D. Molina, R. Benjamins, R. Chatila, and F. Herrera, "Explainable Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information Fusion*, vol. 58, no. October 2019, pp. 82–115, 2020. DOI: 10.1016/j.inffus.2019.12.012.

[16] A. Abdul, J. Vermeulen, D. Wang, B. Y. Lim, and M. Kankanhalli, "Trends and trajectories for explainable, accountable and intelligible systems: An HCI research agenda," in *Proc. of ACM Conf. on Human Factors in Computing Systems (CHI '18)*, Association for Computing Machinery, 2018, pp. 1–18. DOI: 10.1145/3173574.3174156.

[17] N. Magrofuoco, P. Roselli, and J. Vanderdonckt, "Two-dimensional stroke gesture recognition: A survey," *ACM Computing Surveys*, pp. 1–34, 2021. DOI: 10.1145/3465400.

[18] J. Vanderdonckt, P. Roselli, and J. L. Pérez-Medina, "!FTL, an articulation-invariant stroke gesture recognizer with controllable position, scale, and rotation invariances," in *Proc. of the ACM Int. Conf. on Multimodal Interaction (ICMI '18)*, S. K. D'Mello, P. G. Georgiou, S. Scherer, E. M. Provost, M. Soleymani, and M. Worsley, Eds., ACM, 2018, pp. 125–134. DOI: 10.1145/3242969.3243032.

[19] R. Aigner, D. Wigdor, H. Benko, M. Haller, D. Lindlbauer, I. Alexandra, S. Zhao, and J. T. K. V. Koh, "Understanding Mid-Air Hand Gestures: A Study of Human Preferences in Usage of Gesture Types for HCI," *Microsoft Research TechReport MSR-TR-2012-111*, 2012. DOI: 10.1007/978-3-319-59226-8_28.

[20] M. J. Cheok, Z. Omar, and M. H. Jaward, "A review of hand gesture and sign language recognition techniques," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 1, pp. 131–153, 2017. DOI: 10.1007/s13042-017-0705-5.

[21] S. C. Ong and S. Ranganath, "Automatic sign language analysis: A survey and the future beyond lexical meaning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 6, pp. 873–891, 2005, ISSN: 01628828. DOI: 10.1109/TPAMI.2005.112.

[22] A. S. Al-Shamayleh, R. Ahmad, M. A. Abushariah, K. A. Alam, and N. Jomhari, "A systematic literature review on vision based gesture recognition techniques," *Multimedia Tools and Applications*, vol. 77, no. 21, pp. 28 121–28 184, 2018. DOI: 10.1007/s11042-018-5971-z.

[23] M. Yasen and S. Jusoh, "A systematic review on hand gesture recognition techniques, challenges and applications," *PeerJ Computer Science*, vol. 2019, no. 9, pp. 1–30, 2019. DOI: 10.7717/peerj-cs.218.

[24] N. Tubaiz, T. Shanableh, and K. Assaleh, "Glove-Based Continuous Arabic Sign Language Recognition in User-Dependent Mode," *IEEE Transactions on Human-Machine Systems*, vol. 45, no. 4, pp. 526–533, 2015. DOI: 10.1109/THMS.2015.2406692.

[25] M. S. Hasan and H. Yu, "Innovative developments in HCI and future trends," *International Journal of Automation and Computing*, vol. 14, no. 1, pp. 10–20, 2017, ISSN: 17518520. DOI: 10.1007/s11633-016-1039-6.

[26] J. O. Wobbrock, A. D. Wilson, and Y. Li, "Gestures without libraries, toolkits or training: A $1 recognizer for user interface prototypes," *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology*, pp. 159–168, 2007. DOI: 10.1145/1294211.1294238.

[27] L. Anthony and J. O. Wobbrock, "A lightweight multistroke recognizer for user interface prototypes," *Proceedings - Graphics Interface*, pp. 245–252, 2010.

[28] R. D. Vatavu, L. Anthony, and J. O. Wobbrock, "Gestures as point clouds: A $p recognizer for user interface prototypes," *ICMI'12 - Proceedings of the ACM International Conference on Multimodal Interaction*, pp. 273–280, 2012. DOI: 10.1145/2388676.2388732.

[29] R. D. Vatavu, "Improving gesture recognition accuracy on touch screens for users with low vision," *Conference on Human Factors in Computing Systems - Proceedings*, vol. 2017-May, pp. 4667–4679, 2017. DOI: 10.1145/3025453.3025941.

[30] F. M. Caputo, P. Prebianca, A. Carcangiu, L. D. Spano, and A. Giachetti, "Comparing 3D trajectories for simple mid-air gesture recognition," *Computers and Graphics (Pergamon)*, vol. 73, pp. 17–25, 2018, ISSN: 00978493. DOI: 10.1016/j.cag.2018.02.009. [Online]. Available: https://doi.org/10.1016/j.cag.2018.02.009.

[31] M. Ousmer, A. Sluÿters, N. Magrofuoco, P. Roselli, and J. Vanderdonckt, "Recognizing 3D Trajectories as 2D Multi-stroke Gestures," *Proceedings of the ACM on Human-Computer Interaction*, vol. 4, no. ISS, 2020, ISSN: 25730142. DOI: 10.1145/3427326.

[32] S. Kratz and M. Rohs, "A $3 gesture recognizer: Simple gesture recognition for devices equipped with 3D acceleration sensors," *International Conference on Intelligent User Interfaces, Proceedings IUI*, pp. 341–344, 2010. DOI: 10.1145/1719970.1720026.

[33] Y. Li, "Protractor: A fast and accurate gesture recognizer," in *Proc. of the ACM Conf. on Human Factors in Computing Systems*, ser. CHI '10, Atlanta, Georgia, USA: Association for Computing Machinery, 2010, pp. 2169–2172. DOI: 10.1145/1753326.1753654.

[34] S. Kratz and M. Rohs, "Protractor3D: A Closed-Form Solution to Rotation-Invariant 3D Gestures Sven," in *IUI '11: Proceedings of the 16th international conference on Intelligent user interfaces*, 2011, pp. 371–374, ISBN: 9781450304191. DOI: 10.1145/1943403.1943468.

[35] J. Herold and T. F. Stahovich, “The 1\¢ Recognizer: a fast, accurate, and easy-to-implement handwritten gesture recognition technique,” *Proceedings of the International Symposium on Sketch-Based Interfaces and Modeling*, pp. 39–46, 2012. DOI: 10.2312/SBM/SBM12/039-046.

[36] D. Rubine, “Specifying gestures by example,” *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 1991*, vol. 25, pp. 329–337, 1991. DOI: 10.1145/122718.122753.

[37] E.-G. Craciun, I. Rusu, and R.-D. Vatavu, *Free-Hand Gesture Recognizer Pseudocode*, 2016. [Online]. Available: http:%20//www.eed.usv.ro/mintviz/projects/GIVISIMP/data/Pseudocode2.pdf.

[38] G. Saggio, “A novel array of flex sensors for a goniometric glove,” *Sensors and Actuators, A: Physical*, vol. 205, pp. 119–125, 2014, ISSN: 09244247. DOI: 10.1016/j.sna.2013.10.030.

[39] Captiks, *Captiks Official Website*, 2012. [Online]. Available: http://www.captiks.com (visited on 04/28/2020).

[40] J. A. Lester, “Triangles i: Shapes,” *Aequationes mathematicae*, vol. 52, pp. 30–54, 1996. DOI: 10.1007/BF01818325.

[41] L. Luzzi and P. Roselli, “The shape of planar smooth gestures and the convergence of a gesture recognizer,” *Aequationes mathematicae*, 2020. DOI: 10.1007/s00010-020-00712-7.

[42] M. A. Ahmed, B. B. Zaidan, A. A. Zaidan, M. M. Salih, and M. M. B. Lakulu, “A review on systems-based sensory gloves for sign language recognition state of the art between 2007 and 2017,” *Sensors (Switzerland)*, vol. 18, no. 7, 2018, ISSN: 14248220. DOI: 10.3390/s18072208.

[43] A. Maronidis, A. Tefas, and I. Pitas, “Frontal view recognition using spectral clustering and subspace learning methods,” in *Artificial Neural Networks – ICANN 2010*, K. Diamantaras, W. Duch, and L. S. Iliadis, Eds., Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 460–469, ISBN: 978-3-642-15819-3.

[44] S. Zhao, H. Cai, W. Li, Y. Liu, and C. Liu, “Hand Gesture Recognition on a Resource-Limited Interactive Wristband,” *Sensors (Switzerland)*, vol. 21, no. 5713, pp. 1–19, 2021. DOI: https://doi.org/10.3390/s21175713.

[45] V. A. Shanthakumar, C. Peng, J. T. Hansberger, L. Cao, S. C. Meacham, and V. R. Blakely, “Design and evaluation of a hand gesture recognition approach for real-time interactions,” *Multim. Tools Appl.*, vol. 79, no. 25-26, pp. 17 707–17 730, 2020. DOI: 10.1007/s11042-019-08520-1.

[46] J. Nielsen, *Usability Engineering*, 1st ed. Morgan Kaufmann, 1993, ISBN: 0125184050.

Alexandre Calado holds a Biomedical Engineering and Biophysics joint BSc and MSc degree, obtained at the Faculty of Sciences University of Lisbon. He worked for two years as a researcher in Algoritmi Centre, University of Minho, Portugal. Currently he is working on his PhD on Electronics Engineering at the University of Rome “Tor Vergata”. His main research interests include myoelectric prosthetic control, human-machine interaction, sensors, machine learning, rehabilitation, computer vision, signal processing and inclusive technologies.

Paolo Roselli is a Senior Research Associate with tenure in mathematical analysis at Università di Roma “Tor Vergata”, Italy, since 2001. He has been also an invited Professor at Université catholique de Louvain from 2012 to 2020 and at Université Libre de Bruxelles, Belgium for the periods 2014-2015 and 2018-2019. Dr. Roselli received the prize of the Royal Academy of Sciences, Letters and Fine Arts of Belgium in 2014. His research interests include variational methods in Hamiltonian systems, critical point theory, symmetry and symmetry-breaking phenomena, real analysis and fine properties in Lebesgue function spaces, logic, Curry-Howard correspondence and didactic, applications of Clifford geometric algebras to geometric measures, integration and gesture recognition.

Vito Errico graduated in electronic engineering and received the Ph.D. degree in innovative technologies and materials from University of Salento (Italy) in 2008. Since 2013, he has been a Research Fellow at the University of Rome “Tor Vergata,” Rome, Italy. His work involves medical devices and innovative implantable sensors design and development. His research interests are embedded systems, biomedical devices, wireless sensors, and body area networks. Dr. Errico was prized in 2007 in the Marie Curie Contest, and in 2020 as a Young Innovator he received the Innovation Business Award ANGI Prize. He reviews for and is part of the editorial board of indexed peer-reviewed journals.

Nathan Magrofuoco received his Master’s degree and PhD in Computer Science in 2016, resp. 2021, and his Master’s degree in Management Sciences in 2017 with the magna cum laude designation from Université catholique de Louvain, Belgium. He is a member of the Louvain Research Institute in Management and Organizations (LouRIM), Université catholique de Louvain (UCLouvain). His research interests include gesture interaction and multi-context stroke gesture recognition based on Clifford’s algebra.

Jean Vanderdonckt (M’04-SM’08) received an M.Sc. degree in mathematics, an M.Sc. degree in computer science and a Ph.D. degree from the University of Namur, Belgium, in 1987, 1989, and 1997, respectively. He was Visiting Associate Professor at Stanford University in 2000. He is Full Professor at Louvain School of Management, Université catholique de Louvain, Belgium, where he leads the Louvain Interaction Lab. His research interests include human-computer interaction (HCI), engineering interactive computing systems (EICS), intelligent user interfaces (IUI), usability engineering and software engineering. Dr. Vanderdonckt is Associate Editor of ACM Trans. on Interactive Intelligent Systems (TiiS), co-editor-in-chief of the Springer Series of HCI.

Giovanni Saggio (M’91–PhD’97)), researcher (1997-2021), since 2021 Associate Professor with tenure at the University of Rome “Tor Vergata” (Italy), teaching courses of basic and advanced Electronics both for Engineers ((Depts. of Information, Automation, Mathematics, Biomedical, Master of Sound) and health technicians (Dept. of Audiology). GS authored/co-authored 200+ scientific publications, eight patents, several book chapters and is author of nine books about Electronics. He has been working on electronic noise, SAWs, electronic sensors and, in the latest years, on wearable electronics and data analysis for Health. He has been project leader of researches for the Italian Space Agency (ASI), for the Avionic Service of the Italian Defence Department (Armaereo), for the Italian Workers’ Compensation Authority (INAIL), currently member of the Italian Space Biomedicine, and advisor for Research and Innovation to the Italian Deputy Minister of Health