UQAC

Université du Québec
à Chicoutimi

Towards streaming gesture recognition

Baptiste Lemarcis

Mémoire présentée à l'Université du Québec à Chicoutimi en vue de l'obtention du grade de la Maîtrise (M. sc.) en informatique profil recherche

Québec, Canada

ABSTRACT

The emergence of low-cost sensors allows more devices to be equipped with various types of sensors. In this way, mobile device such as smartphones or smartwatches now may contain accelerometers, gyroscopes, etc. This offers new possibilities for interacting with the environment and benefits would come to exploit these sensors. As a consequence, the literature on gesture recognition systems that employ such sensors grow considerably. The literature regarding online gesture recognition counts many methods based on Dynamic Time Warping (DTW). However, this method was demonstrated has non-efficient for time series from inertial sensors unit as a lot of noise is present. In this way new methods based on LCSS (Longest Common SubSequence) were introduced. Nevertheless, none of them focus on a class optimization process.

In this master thesis, we present and evaluate a new algorithm for online gesture recognition in noisy streams. This technique relies upon the LM-WLCSS (Limited Memory and Warping LCSS) algorithm that has demonstrated its efficiency on gesture recognition. This new method involves a quantization step (via the K-Means clustering algorithm) that transforms new data to a finite set. In this way, each new sample can be compared to several templates (one per class). Gestures are rejected based on a previously trained rejection threshold. Thereafter, an algorithm, called SearchMax, find a local maximum within a sliding window and output whether or not the gesture has been recognized. In order to resolve conflicts that may occur, another classifier (i.e. C4.5) could be completed. As the K-Means clustering algorithm needs to be initialized with the number of clusters to create, we also introduce a straightforward optimization process. Such an operation also optimizes the window size for the SearchMax algorithm. In order to demonstrate the robustness of our algorithm, an experiment has been performed over two different data sets. However, results on tested data sets are only accurate when training data are used as test data. This may be due to the fact that the method is in an overlearning state.

RÉSUMÉ

L'apparition de nouveaux capteurs à bas prix a permis d'en équiper dans beaucoup plus d'appareils. En effet, dans les appareils mobiles tels que les téléphones et les montres intelligentes nous retrouvons des accéléromètres, gyroscopes, etc. Ces capteurs présents dans notre vie quotidienne offrent de toutes nouvelles possibilités en matière d'interaction avec notre environnement et il serait avantageux de les utiliser. Cela a eu pour conséquence une augmentation considérable du nombre de recherches dans le domaine de reconnaissance de geste basé sur ce type de capteur. La littérature concernant la reconnaissance de gestes en ligne comptabilise beaucoup de méthodes qui se basent sur Dynamic Time Warping (DTW). Cependant, il a été démontré que cette méthode se révèle inefficace en ce qui concerne les séries temporelles provenant d'une centrale à inertie puisqu'elles contiennent beaucoup de bruit. En ce sens de nouvelles méthodes basées sur LCSS (Longest Common SubSequence) sont apparues. Néanmoins, aucune d'entre elles ne s'est focalisée sur un processus d'optimisation par class.

Ce mémoire de maîtrise consiste en une présentation et une évaluation d'un nouvel algorithme pour la reconnaissance de geste en ligne avec des données bruitées. Cette technique repose sur l'algorithme LM-WLCSS (Limited Memory and Warping LCSS) qui a d'ores et déjà démontré son efficacité quant à la reconnaissance de geste. Cette nouvelle méthode est donc composée d'une étape dite de quantification (grâce à l'algorithme de regroupement K-Means) qui se charge de convertir les nouvelles données entrantes vers un ensemble de données fini. Chaque nouvelle donnée peut donc être comparée à plusieurs motifs (un par classe) et un geste est reconnu dès lors que son score dépasse un seuil préalablement entraîné. Puis, un autre algorithme appelé SearchMax se charge de trouver un maximum local au sein d'une fenêtre glissant afin de préciser si oui ou non un geste a été reconnu. Cependant des conflits peuvent survenir et en ce sens un autre classifieur (c.-à-d. C4.5) est chainé. Étant donné que l'algorithme de regroupement K-Means a besoin d'une valeur pour le nombre de regroupements à faire, nous introduisons également une technique simple d'optimisation à ce sujet. Cette partie d'optimisation se charge également de trouver la meilleure taille de fenêtre possible pour l'algorithme SearchMax. Afin de démontrer l'efficacité et la robustesse de notre algorithme, nous l'avons testé sur deux ensembles de données différents. Cependant, les résultats sur les ensembles de données testées n'étaient bons que lorsque les données d'entrainement étaient utilisées en tant que données de test. Cela peut être dû au fait que la méthode est dans un état de surapprentissage.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF EQUATIONS

LIST OF ACRONYMS

9DOF:	9 Degree Of Freedom
DNA:	DeoxyriboNucleic Acid
FN:	False Negative
EMG:	ElectroMyoGram
HMM:	Hidden Markov Model
DTW:	Dynamic Time Warping
IOT:	Internet Of Thing
IMU:	Inertial Measurement Unit
LCSS:	Longest Common SubSequence
LIARA:	Laboratoire d'Intelligence Ambiante pour la Reconnaissance d'Activité
LM-WLCSS:	Limited Memory and Warping Longest Common Sub-Sequence
MCU:	MicroController Unit
MEMS:	Micro-Electro-Mechanical Systems
OLM-WLCSS:	Optimized Limited Memory and Warping Longest Common Sub-Sequence
OS:	Operating System
RAM:	Random Access Memory
TMM:	Template Matching Method
Hz:	Hertz
FP:	False Positive
TN:	True Negative
WLCSS:	Warping Longest Common SubSequence
TV:	TeleVision
TP:	True Positive
Wi-Fi:	Wireless Fidelity
WiiMote:	Wii Remote controller

ACKNOWLEDGEMENTS

CHAPTER 1

INTRODUCTION

1.1   RESEARCH CONTEXT

Nowadays, with the wide spread of computers and smartphones, traditional communication channels are likely to be keyboards and mouses. But it is not a natural way to communicate, as Humans tend to primarily communicate by speaking. Moreover, research proves that a large part of information is conveyed through gestures (Burgoon, Guerrero, & Floyd, 2016). In social interaction humans tend to carry information, thanks to their body language and more particularly with hand gestures. In this way, gestures can be considered as a natural way of communication. With the appearance of 3D virtual environments, keyboards and mouses were reviewed as an ineffective communication channel. Moreover, when considering possible benefits that gesture recognition would bring in computer interaction, the interest in hand gestures recognition systems has increased. In the literature researches on gesture recognition tend to build an interface that could recognize gestures performed by a user. Applications for these systems are multiple as they can translate sign language (Pan *et al.*, 2016; Rung-Huei & Ming, 1998), or control an application like the Myo armband (Thalmic Labs Inc) does.

As explained by N. H. A.-Q. Dardas (2012), hand gestures are composed of two distinct characteristics, the position (posture) and movements (gestures) that are both

crucial information in a human computer interaction context. However, to recognize those characteristics the posture and the gesture must be modeled in a spatial and temporal way. Gestures recognition methods can be divided based on their techniques such as: vision, gloves, colored markers, etc (Chaudhary, Raheja, Das, & Raheja, 2013; Ibraheem & Khan, 2012). Considering that each method as weakness and strength, and let us briefly explain vision approaches ones as this is the most common approach of the literature. Human-computer interfaces relying on vision try to be close to an eye as a human will mostly recognize a gesture thanks to his vision (N. H. A.-Q. Dardas, 2012). Thus, users do not wear any device, they only executes gestures as normal. Therefore the ease and naturalness of the interaction are preserved. However, it implies many problems as the user has to always be recorded; any occlusion problem is detrimental for the system. Moreover, the system as to be tolerant with background changes, light condition, it cannot be forced to a specific environment, the hand has to be tracked and its posture to be determined. These challenges are specific to vision-based approaches, as for example an approach relying on data from a glove is not affected by cameras problems but may constrain the movement.

More recently, the emergence of low-cost MEMS (Micro-Electro-Mechanical Systems as accelerometers, magnetometers, etc.) technology brings new sensors in everyday life devices as in smartphones or smartwatches (Guiry, van de Ven, & Nelson, 2014). Thus, new possibilities for interaction with our environment appear and the traditional way of communication, (based on a keyboard), tends to evolve to a

2

gesture-based system. Indeed, with appropriate small and wireless devices in clothes and appropriate techniques we could recognize gestures performed by the user and control home appliances or provide helped in some activities (Akl, Feng, & Valaee, 2011). Compared to the two previous approaches, this one does not decrease the naturalness of the interaction like with a data-glove approach and does not need to be as constraint as the vision based methods.

1.2   GESTURE RECOGNITION

The human body is in constant movement and whether its eyes, arm, face or hands these motions could be useful (Rizwan, Rahmat-Samii, & Ukkonen, 2015). Indeed, gestures are present in everyday communications to convey a large part of information, and when we interact with the environment. A movement of a body part involves two characteristics (Akl *et al.*, 2011). First, the posture that is the static position. It does not include the movement. Second, the motion itself that corresponds to the dynamic movement of the body part. However, for a given gesture there are many possible representations depending on the individual, the context and even the culture. For example, in France the number two is represented with the forefinger and the middle finger representing an insult in England. Moreover, in some country the head movement for an affirmative or a negative response is reverse. Furthermore, the same individual will vary his gesture over multiple instances.

Movements of the human body can be understood and classified thanks to a process called gesture recognition. However, as hand gestures are considered as the most natural and expressive way of communication, they are the most used. Gesture recognition has become important in a wide variety of applications such as gesture-to-speech in sign languages (Kılıboz & Güdükbay, 2015; Rung-Huei & Ming, 1998), in human computer interaction (Song, Demirdjian, & Davis, 2012) and even in virtual reality (Y. Liu, Yin, & Zhang, 2012). In fact, gesture recognition can really be useful as recognize gesture of a hearing impaired could facilitate the communication as it could be possible to translate sign language. Another application is helping people in rehabilitation, with proper sensors such as inertial sensors the movement could be detected and a success rate could be computed. Gesture recognition could also replace the traditional communication channel between a human and a computer, by replacing some mouse and keyboard interaction by a gesture. In virtual reality, gesture recognition could be implemented to increase the immersion of the player in an environment. As a result, it appears that some benefits could certainly come from exploiting gesture recognition.

1.3   ACQUISITION METHODS

Gesture recognition starts with sensing human body position, configuration (angles and rotation), and movement (velocities or accelerations). The process of sensing can be done via specialized devices attached to the user, as inertial measure units (accelerometer, magnetometer, etc.), gloves, clothes with integrated sensors or

even cameras with the appropriate techniques (Mitra & Acharya, 2007). However, each technology as its weakness as the accuracy, user comfort, cost, latency, etc (Akl *et al.*, 2011). For example, gestures interface relying on gloves requires a load of cables connected to a computer that decreases the ease and naturalness of the interaction between the user and a computer. On the other hand, vision-based techniques overcome this problem but are sensitive to the occlusion of part of the user body. However, vision-based techniques are the most present in literature for gesture recognition (Rautaray & Agrawal, 2015). Gesture recognition methods based on computer vision techniques vary according to some criteria as: the number of cameras, their speed and latency, environment (lightning), the speed of the movement, restrictions on clothing (no green shirt with a green background), features (edges, regions, silhouettes, etc.), and whether the technique is based on 2D or 3D. But these constraints limit the applications of vision-based techniques in a smart environment. Indeed, as illustrated in Akl *et al.* (2011), supposing the user is at home and has a vision-based system to detect some gestures to interact with the TV (TeleVision). If the user performs the gesture to increase the volume while all the lights are off, the gesture recognition system will have difficulties because of the poor lighting condition. One possible way to overcome such issue is to use a really more expensive camera with night vision. As well it would be unnatural and uncomfortable to stand up and face to the camera in order to execute a gesture.

In order to recognize gestures, another alternative is to sense gestures with other techniques such as the ones based on IMU or electromyogram (EMG). The

application domain application for each of these techniques differs. Indeed, an accelerometer-based technique is well suited for large hand movements, nevertheless it will not be able to detect the movement of the finger, while the EMG-based technique is sensitive to muscle activation and therefore will detect when a finger move. However, recognize finger gestures with an EMG are difficult due to some reproducibility and discriminability problem. Only 4 to 8 hand gestures can be easily identified with an EMG and therefore this limit the possible actions (Akl *et al.*, 2011). Thereby, after studying means of acquisition in literature, an inertial measurement unit is chosen to be the sensing devices to acquire necessary data for gesture recognition. In the last decade, thanks to the emergence of low-cost MEMS technology, number of techniques for gesture recognition based on IMU (or just accelerometer) increase. As a matter of fact, a lot of these sensors are now embedded in most of the everyday life object as smartphones, smart watch or smart bracelets (Shoaib, Bosch, Incel, Scholten, & Havinga, 2015). Therefore, new possibilities in terms of applications appear such as sports tracking or video games.
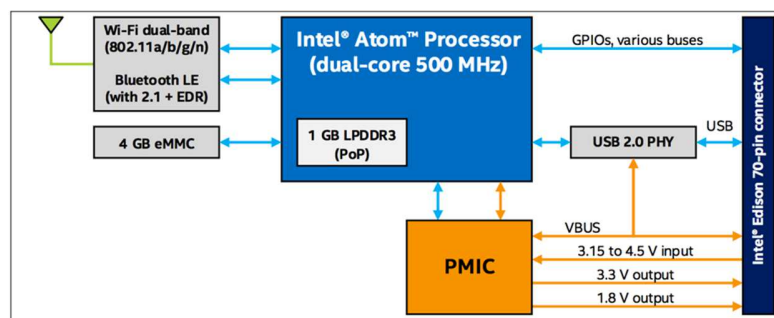
Figure 1: (Source: Intel® Edison Compute Module. 2016, September 7 in Intel® Website).

6
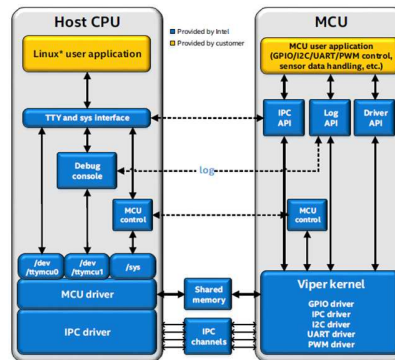
Figure 2: (Source: Using an MCU on the Intel® Edison Board with the Ultrasonic Range Sensor. 2016, September 7 in Intel® Website).

1.3.1 INTEL EDISON

The Intel® Edison is an Internet Of Thing (IOT) board from Intel®, designed to provide an easy way for prototyping or commercial ventures. Figure 1 illustrates the Intel® Edison. This board is composed of a dual-core Intel® Atom processor clocked at 500Mhz and 1 Gigabyte of Random Access Memory (RAM), allowing running multiple applications. In addition to the processor, the Intel® Edison contains a MicroController Unit (MCU) clocked at 100Mhz, illustrates in Figure 2. The MCU allows the user to benefits of real-time and power efficiency that can be required to fetch sensors. Indeed, as the MCU is connected to the 70-pin connector of the Intel® Edison, the user could run a fetching program that requires a complex management of time and by transitivity a real time Operating System (OS). Then, an application on the embedded Linux running on the processor could process data fetched by the MCU.

Figure 3: 9Dof Block (Left), Battery Block (Middle), Base block (Right).

Thanks to its integrated wireless connection (Wi-Fi and Bluetooth), the Intel® Edison can rapidly transfer sensed data to a computer. Moreover, the Intel® Edison is powerful enough to run some gesture recognition algorithms. However, the board itself does not include sensors, but the company Sparkfun create a whole range of "block" that easily plug on the "base block" where the Intel® Edison is. In this way it is easy to build prototypes with 9 Degrees Of Freedom (9DOF) inertial measurement unit (accelerometer, gyroscope and magnetometers) and a battery. Figure 3 shows a 9DOF, a battery and a base block with the Intel® Edison. In this configuration, we attach the LSM9DS0 IMU that combines a 3-axis accelerometer, a 3-axis gyroscope and a 3-axis magnetometer that is connected *via* the I2C bus of the Intel Edison. Each sensor of the IMU supports a lot of range, the accelerometer scale can be set to $\pm$ 2, 4, 6, 8 or 16g, the gyroscope supports $\pm$ 245, 500 and 2000 °/s and the magnetometer as a scale range of $\pm$ 2, 4, 8 or 12 gauss.

1.3.2 THE IMPORTANCE OF A 9 DEGREE OF FREEDOM

Accelerometers are devices for measuring the acceleration of moving objects. Figure 4 and Figure 5 illustrate raw acceleration waveforms of two instances of the gesture OpenWaterReservoirLid. It appears that in two instances of the same gesture, the accelerometer data are not likely to be the same. Indeed, tilting an accelerometer result in different data even if the gesture performed by the user is the same. Other sensing devices as the gyroscope or the magnetometer can be added to the accelerometer to provide more information about the gesture. The gyroscope is a device that allows the calculation of orientation and rotation. Figure 6 and Figure 7 illustrate raw rotation waveforms of two instances of the gesture OpenWaterReservoirLid. In the LSM9DS0, the magnetometer measures magnetic fields and can be used as a compass. Figure 8 and Figure 9 illustrates raw magnetic field waveforms of two instances of the gesture OpenWaterReservoirLid.
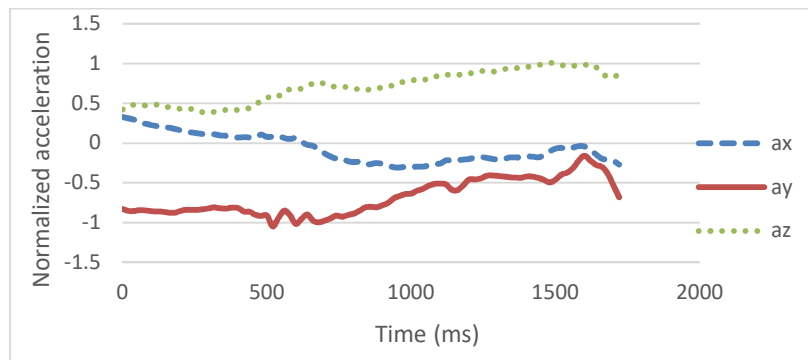
Figure 4: Accelerometer data for first instance of OpenWaterReservoirLid.
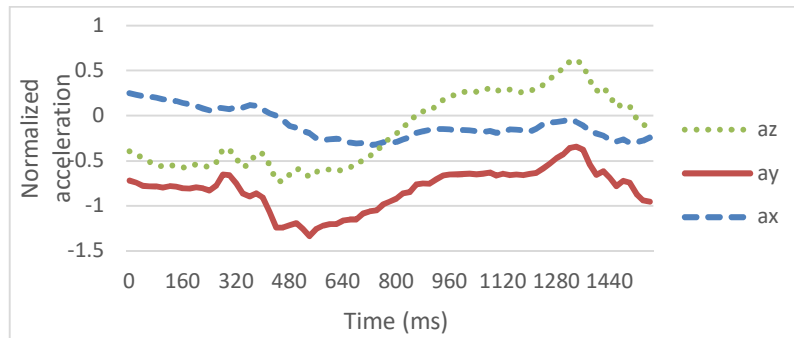
Figure 5: Accelerometer data for second instance of OpenWaterReservoirLid.
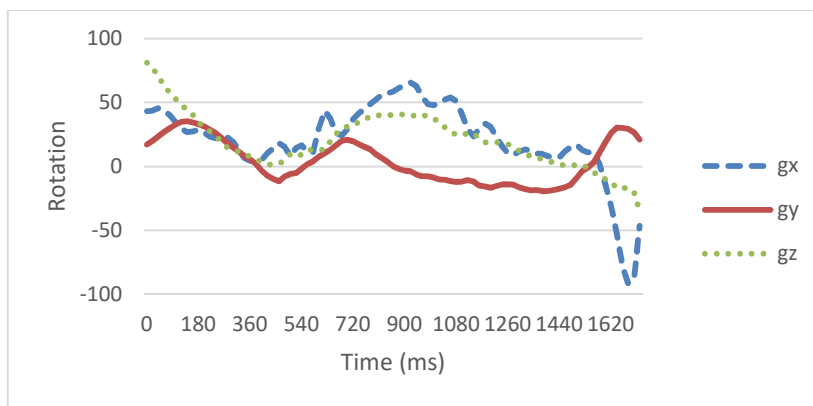
Figure 6: Gyroscope data for first instance of OpenWaterReservoirLid.
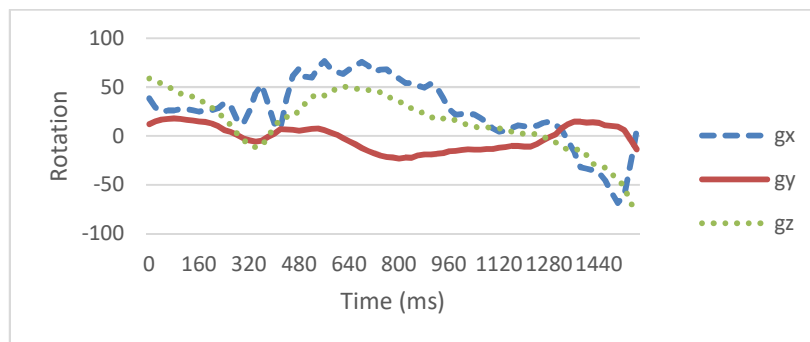
Figure 7: Gyroscope data for second instance of OpenWaterReservoirLid.
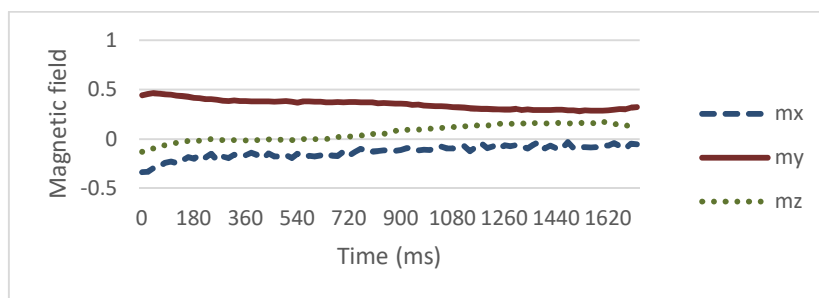
Figure 8: Magnetometer data for first instance of OpenWaterReservoirLid.
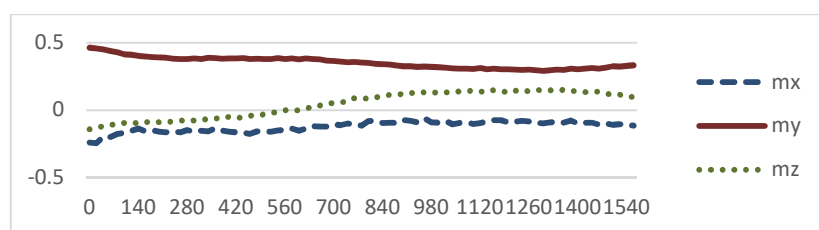
Figure 9: Magnetometer data for second instance of OpenWaterReservoirLid.

1.4 CONTRIBUTION OF THIS THESIS

The literature regarding online gesture counts many methods such as Hidden Markov Model (Hyeon-Kyu & Kim, 1999), Support Vector Machine (N. H. Dardas & Georganas, 2011) and Template Matching Methods (TMMs). TMMs express gestures as templates that are compared with the data stream afterward. The objective of such a computation is to find similarities, where the highest affinity involves the recognition of the fittest gesture. To do so, TMMs may employ Dynamic Time Warping (DTW) as similarity measure (Reyes, Dominguez, & Escalera, 2011).

Although DTW-based TMMs achieve accurate results, the work described in (Vlachos, Hadjieleftheriou, Gunopulos, & Keogh, 2003) shows that this method is not well suited to handle time series and noise produced by inertial sensors. In that sense, the LM-WLCSS (Limited Memory and Warping Longest Common Sub-Sequence) aims at overcoming issues brought by DTW. This method relies upon the WLCSS method (Long-Van, Roggen, Calatroni, & Troster, 2012), an extension of the LCSS problem. However, Roggen, Cuspinera, Pombo, Ali and Nguyen-Dinh (2015) did not focus on class optimization and set arbitrary parameters for the clustering algorithm and windows size. In this thesis, we present a new method based on the LM-WLCSS and focus on the class optimization process to spot gestures of a stream. This in a purpose of trying to improve the LM-WLCSS algorithm. To achieve it, we train and optimize the LM-WLCSS algorithm for each class. More precisely, the process that convert the uncountable set of accelerometer data to a countable one, called the quantization process, is performed for each gesture independently as the entire recognition flow. The final decision is achieved through a decision fusion.

1.5   RESEARCH METHODOLOGY

Gestures are parts of our language, we move every day to speak, walk, for almost everything. In this way, gesture recognition become a new research area as benefits would certainly come from exploiting them. However, gesture recognition brings some challenge as recognize a gesture during its performance, correctly delimit the start and the end of a gesture, the multi-gesture problem, etc. For this

master thesis we wanted to improve a gesture recognition technique to resolve a maximum of these challenges. To achieve this we divided our project in four distinct phases.

The first phase was to gain knowledge for the targeted domain of research *via* a review of the literature on online gesture recognition (N. H. Dardas & Georganas, 2011; Hartmann & Link, 2010; Hyeon-Kyu & Kim, 1999). In particular, the project was focused on methods based on the LCSS problem (Hirschberg, 1977) and a study was performed to understand it. This has provided an overview of the gesture recognition techniques. It has also helped to understand how to bring these methods in Smarthome to assist people with reduced autonomy. Moreover, a state of the art was aimed at existing gesture recognition method. This state of the art has brought possible solutions leading to the contribution of this thesis.

The second phase consisted of the optimization of an existing gesture recognition technique by providing new theoretical basis to solve the issues introduced in the earlier sections. To do this, an improvement of the Limited-Memory and WarpingLCSS (LM-WLCSS) has been decided. In fact, this method has proven to be reliable with noisy signals and show great results on data sets.

This third phase for this project was to make a software implementation of this new theoretical basis to validate it and to provide comparison elements for other gesture recognition techniques. This implementation was developed with the

programming language C# from Microsoft and was run on the Workstation of the LIARA laboratory.

The last phase dwells in the validation of the new implemented method. The first step was to construct the scenario used in the testing step. For this project, the well-known *MakeCoffee* activity was chosen. However, as the new method is for gesture recognition, this activity was represented as a sequence of 14 gestures. The second step was to assemble the sensors (Intel® Edison) with batteries and Wi-Fi board. Results and further details will be provided in Chapter 3.

1.6   THESIS ORGANISATION

This thesis is organized into 4 chapters. The first chapter that is ending consisted into an introduction of the research project. In this way we first described our context for this study and issues that are raised in the literature. This part allows understanding problems in gesture recognition system and bringing examples to illustrate the importance of a 9 Degree of Freedom sensor.

The second chapter provides an introduction to one of the most common methods employs in gesture recognition systems, DTW, and the LCSS problem which our method is based on. Then, a review of current existing approaches in our field of research takes place. First we will focus on the presentation of some methods based on the distance measure DTW to understand problems that this method raised.

In a second time, techniques based on the LCSS are introduced to reveal limitations of this work. This chapter will conclude with an evaluation of these reviews to better understand our contributions.

The third chapter details the proposed systems of this master thesis. The first part of this chapter is about the theoretical definition of this system and how we modify the LM-WLCSS method. In a second time we examine the practical definition by showing our implementation for the following evaluation. The next section is a formal description on which data set is employed for the validation of the method, which metrics are used and results obtained. The final part of this chapter concludes by offering a summary of the introduced method and its performance.

Finally, the fourth and final chapter draws a general conclusion of this master thesis project by starting with a brief summary of previous chapter. Then each step of the methodology is reviewed to show how it was achieved. This chapter concludes with a personal assessment of this first experience as a scientific researcher.

CHAPTER 2

STATE OF THE ART

Due to its involvement in many human-computer interactions, some techniques such as computer vision-based (Rautaray & Agrawal, 2015), data-glove based (Kim, Thang, & Kim, 2009), inertial sensors (Long-Van *et al.*, 2012), etc. were employed in gesture recognition. With the emergence of MEMS on smart objects (smartphone, smarwatch, etc.) we review in this state of the art inertial sensor-based gesture recognition methods. For a more detailed analysis of other techniques we may refer to (Ibraheem & Khan, 2012; Mitra & Acharya, 2007).

With the emergence of low-cost MEMS technology, the number of systems relying on inertial measurement units or a single accelerometer tends to increase. The literature shows that many methods already exist and are based on various techniques as DTW and HMM (Jang, Han, Kim, & Yang, 2011; J. Liu, Zhong, Wickramasuriya, & Vasudevan, 2009; Pylvänäinen, 2005; Schlömer, Poppinga, Henze, & Boll, 2008). However, more recently new methods explore the viability of the LCSS problem in accelerometer based gesture recognition systems (Long-Van *et al.*, 2012).

In this section we introduce one of the most common methods employs in gesture recognition systems, DTW, and the LCSS problem which our method is based on. Then, a review of current existing approaches in our field of research takes place. First we will focus on the presentation of some methods based on the distance

measure DTW to understand problems that this method raised. In a second time, techniques based on the LCSS are introduced to reveal limitations of this work. This chapter will conclude with an evaluation of these reviews to better understand our contributions.

2.1   DTW

The Dynamic Time Warping (DTW) algorithm (Berndt & Clifford, 1994; Müller, 2007) was introduced to compare two time series. Unlike the Euclidean distance, this algorithm can measure the similarity between two sequences regardless the size of each of them. This particularity leads to a more frequent usage of DTW over the Euclidean distance.

Let define S_1 and S_2 two sequences (or time series) of respective N and M size, where:

$$S_1 = \alpha_1, \alpha_2, \dots, \alpha_n, \dots, \alpha_N \text{ with } \alpha_n \in S, \text{ for n} \in [1:N]$$

$$S_2 = \beta_1, \beta_2, \dots, \beta_m, \dots, \beta_M \text{ with } \beta_m \in S, \text{ for m} \in [1:M]$$

To compare two different elements α and β of the sequence, one needs a local cost (or distance) measure. Let denote the computation of this measure by a function λ as many distance measures exists, define as follows:

$$\lambda : S{\times}S \rightarrow \mathbb{R}_+$$

Equation 1 : The distance function between elements of two time series

The comparison of the sequences S_1 and S_2 start with the cost calculation of each pair (α, β), obtaining the $N{\times}M$ cost matrix Λ defined by $\Lambda(n, m) = \lambda(\alpha_n, \beta_m)$. Then, the goal is to find a path, called warping path (W), in this matrix that will represent the similarity of S_1 and S_2. A warping path W is defined as a sequence, where each element corresponds to an association of a α_n and a β_m. The lth element of W is defined as $w_l = (n_l, m_l) \in [1{:}N]{\times}[1{:}M]$ for $l \in [1{:}L]$.

$$W = w_1, w_2, \dots, w_l, \dots, w_L \text{ with } \max(n, m) \leq L \leq m + n - 1$$

Equation 2 : Warping path definition
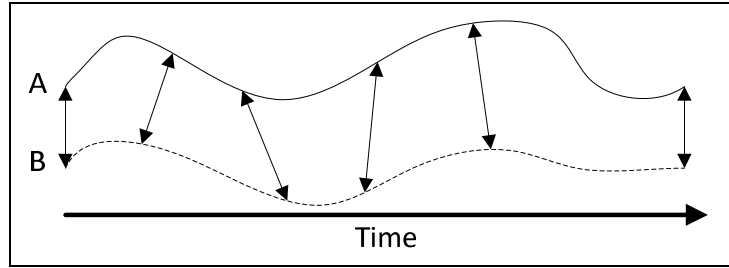
Figure 10 : Two time series with the representation of the warping path indicated by arrows.

However, it is impossible to find a warping path over all the possibilities, their number is far too high. To reduce the search space some constraints have to be followed. First, a boundary condition requires the starting and ending points of the path to be respectively the first and last pair of elements (i.e. $w_1 = (1,1)$ and $w_L = (n, m)$). The second constraint is the monotonicity involving the respect of the time

18

order: for each $w_l = (n_l, m_l)$ and $w_{l-1} = (n_{l-1}, m_{l-1})$, $n_l \geq n_{l-1}$ and $m_l \geq m_{l-1}$. The last condition is the continuity (or step size), no value can be skipped (i.e. $n_l - n_{l-1} \leq 1$ and $m_l - m_{l-1} \leq 1$). The resulting space still contains many warping path, however, only the one with the minimal total cost is considered as optimal. The minimum cost matrix can be computed thanks to the following dynamic programming formulation, where $\gamma(n, m)$ is the minimum cumulative cost for the pair (n, m).

$$\gamma(n, m) = \lambda(n, m) + \min[\gamma(n-1, m), \gamma(n, m-1), \gamma(n-1, m-1)]$$

Equation 3 :Dynamic programming formulation for matrix of cost

The cumulative cost is computed with, a sum between the cost of the current element of the matrix, and the minimum cumulative distance (cost) of its predecessor neighbors. Due to the recursive aspect, the last value $\gamma(N, M)$ represent the lowest cost for a warping path and allow an easy backtracking of it. The dynamic time warping algorithm is formally defined as:

$$DTW(S_1, S_2) = \gamma(N, M)$$

Equation 4 : Definition of the similarity between two time series.

2.1.1 EXAMPLE OF DTW

19

To help better understanding the concept previously described, an example of how works DTW to compute similarity between two time-series A and B is given here. Let define S_1, S_2 and the cost function as follows:

$$S_1 = [8,8,10,10,10,12,12,13]$$

$$S_2 = [8,10,12,13]$$

$$\lambda(i,j) = (S_{1_i} - S_{2_j})^2$$

In this case the value of N and M are respectively 8 and 4, the distance matrix will be 8×4. This matrix illustrates on the left of the Table 1, is constructed from the distance function as previously described, so the (i,j)-th element of the matrix is $\lambda(i,j) = (S_{1_i} - S_{2_j})^2$. Once this step is done, the minimified cost matrix can be computed from the distance one by applying the Equation 3. Then, the similarity cost of the two segments is the one in the top right corner of the matrix. It associates warping path can be backtracked relying on the previous minimum cost. Here the similarity cost is 0 as the sequence S_2 is a compression of S_1, the warping path is identified by the green color.

Table 1: The cost matrix (left) and the minified cost matrix (right).

13	25	9	1	0
12	16	4	0	1
12	16	4	0	1
10	4	0	4	9
10	4	0	4	9
10	4	0	4	9
8	0	4	16	25

69	17	1	0
44	8	0	1
28	4	0	1
12	0	4	13
8	0	4	13
4	0	4	13
0	4	20	45

8	0	4	16	25

| | 8 | 10 | 12 | 13 |

0	4	20	45

2.1.2 DTW-BASED METHODS

Akl and Valaee (2010) introduce a new method for gesture recognition based on DTW. In order to sense gestures of the users a Nintendo Wii Remote controller (or WiiMote) was held by the user and thanks to its integrated 3-D accelerometer data from the gesture can be saved. Boundaries of each gesture are well defined as the user press and hold the "B button" of the controller while performing the given gesture. To improve recognition rates and computational cost of DTW, a temporal compression (Akl & Valaee, 2010) is applied as a pre-processing to remove data that are not intrinsic to the gesture. This phase is performed thanks to a sliding window of 70ms with a 30ms step. Akl and Valaee (2010) compare their method in a user-dependent and user-independent case, and thus the model for each case is different. Let understand the user-dependent model as the user-independent one takes some of its component from this one.

The training phase (were the model is build) of this model starts by the temporal compression, thus all minor tilting or hand-shaking effect will be removed from the signal. Then, DTW constructs the similarity matrix by comparing the similarity of each pair of M randomly choose gestures. This matrix is then processed by a clustering algorithm that will divide it into N (number of gestures) clusters. In this method an Affinity Propagation (Frey & Dueck, 2007) was chosen over a K-Means (Hartigan, 1975) because the Affinity propagation consider all data as

exemplars and recursively transmits real-valued messages until a good set of exemplars and clusters emerge. Resulting into N clusters each identified with an exemplar. In the case of user-independent, gestures for the similarity matrix is chosen between user and thus a number a $N \times K$ (with K less than the number of users). Then Affinity propagation tries to create a cluster for each gesture as for the user-dependent, however, it does not always succeed and thus a gesture can be in multiple clusters but all repetition of a given gesture and user are in the same clusters. The output of this training is an arbitrary number of exemplar.

Exemplars from the training phase are stored for the testing phase (where we validate the method), also different between user-dependent and user-independent cases. First, in the user-dependent case the incoming signal is still temporally compressed before it is compared to exemplars thanks to DTW. An unknown gesture is classified based on its lowest cost with exemplars. In order to examine the dependence of the amount of training repetitions the parameter M was varied and as a result more training repetitions yield to a better performance. In the case of user-independent recognition, the way of recovering gesture change as multiple gestures can fall into the lowest cost cluster. To overcome this issue all exemplars of these clusters are recovered and the one with the highest similarity. For the test in a user-independent case they randomly choose 3 users (K=3). Performance for this new method is promising as for a user-dependent system the accuracy is up to 100% with the proper amount of training repetition. For the user-independent, the accuracy is lower with a maximum of 96% when the system as to only recognize 8 of the 18 gestures and a minimum of 90% with all the gestures, still competitive with other methods. However, to create data sets Akl and Valaee (2010) ask for their users to try

their best in not tilting the accelerometer while performing gesture and hold the button only during the gestures. This leads into a near perfect usage case, as in real world a gesture recognition system based on an accelerometer will always run and therefore a lot of noise will be presented and this method could not be that effective.

Choe, Min and Cho (2010) present a new method for gesture recognition on a mobile phone. This new algorithm employs the DTW method in a K-Means clustering method. More precisely the first step is a pre-processing that will reduce noise produce by the accelerometer. It consists of the segmentation of the input sequence based on the mean variation and the maximum values within a sliding window of 120ms with steps of 60ms. Moreover, segmented gestures shorter than a defined minimum length is considered as noise. Then a quantization and smoothing step occurs by averaging sequence within the sliding window. To reduce additional effects related to gravity; g is subtracted from the input sequence. The next step is to elect a template in order to recognize gesture, and because of the dynamics of input gesture various patterns are needed. These templates are chosen from the whole training set and K-Means offer great performance to do it. However, the K-Means clustering algorithm based on the Euclidean distance takes vector of the same length as input, which is not possible with acceleration data. In order to overcome this problem Choe *et al.* (2010) replace the Euclidean distance with DTW as this algorithm respects the time series. The gesture matching method is then tested on a mobile phone with 20 gestures that are considered as recurrent while browsing mobile content. The internal accelerometer sends data at 50 Hz and is initialized thanks to a button. Then the method automatically detects start and end point of a

gesture. Moreover, the user can add gestures as long as some instance of it is recorded. For evaluation purposes this algorithm was also implemented and tested on a computer. In this case four methods of template elections are compared. First each instance of the whole training set is chosen as a template (All). Secondly, the random k (Ran k) that chooses k random templates over the training set. The third method is Euclidean k (Euc k) that also choose k templates but is based on the Euclidean distance, instances of the training set were resize for this method. Last method is the one that Choe *et al.* (2010) introduce, the same as Euc k but with DTW (DTW k). Tests were performed with $k = 3$ and $k = 5$. The resulting measures show that the accuracy of the DTW5 and All method is pretty similar and higher than other methods. Moreover, the DTW5 method offers a higher execution speed than all cases (~400ms against 75ms for the DTW5). This new method proves that it works well on simple gesture used for mobile browsing content but not necessarily with more complex gesture.

2.2 LCSS

In Biology, researchers often need to match two or more organisms by comparing their deoxyribonucleic acid (DNA). This consists in studying strand of DNA, composed of bases (sequence of molecules). A base is either adenine, quinine, cytosine or thymine and representing a strand of DNA by the finite set compose of base initial letters give a string. Let define two strands of DNA S_1, S_2 as follows:

$$S_1 = [ACGTGGTTACCAATGTC]$$

$$S_2 = [GTAACTACATGCAA]$$

The reason to compare these two strands is to measure their similarity; a high one implies the two organisms are likely to be the same. To determine it, many ways exist and as the DNA can be represented with strings, one solution is to compare the associated strings and identified their eventual likeliness. For example, to determine the similarity between two strings, one can verify if one is a substring of the other. However, in our case none of the two strings is a substring of the other one. Another way is to represent the similarity by the number of changes to get the second DNA strand from the first. One final solution is to find a third string S_3 (or strand) that represents S_1 and S_2. A valid representation is a string where each element is in both S_1 and S_2. Based on this new strand must be in the same order as they appear in S_1 and S_2, however the sequence can be discontinued. In this way, the size still represents the similarity and the longer the strand is the higher is the similarity. For our example the longest common sequence S_3 is:

$$S_3 = [AACCBAC]$$

Figure 11 : Representation of the LCSS.

2.2.1 LCSS EXAMPLE

This problem is known in literature to be the longest common subsequence problem (LCSS). We review a subsequence of a given sequence as this sequence private of one or more of its elements. In other words, let $S_1 = [\alpha_1, \alpha_2, \dots, \alpha_n, \dots, \alpha_N]$ and $S_2 = [\beta_1, \beta_2, \dots, \beta_m, \dots, \beta_M]$ be two sequences, S_2 is a subsequence of S_1 if a consecutive part of S_1 represent the entire sequence S_2. For example, let's define these two sequences as follows:

$$S_1 = [ACCCGGTT\,ACGT\,AAA]$$

$$S_2 = [ACGT]$$

In our example, the entire sequence S_2 is in S_1 and as it is previously defined, if a string represents a consecutive part of another string that means the first one is a subsequence of the second. Then, S_2 is a subsequence of S_1. Another possibility is that S_2 is a common subsequence of two given strings. To be a common subsequence, the string S_2 needs to be a subsequence of two strings. Let modify our example to illustrate it:

$$S_1 = [ACCCGGTT\,ACGT\,AAA]$$

$$S_2 = [ACGT]$$

$$S_3 = [AAGGT\,ACGT\,CAG]$$

For this new example, the sequence S_2 is a subsequence of both S_1 and S_3; S_2 is a common subsequence of S_1 and S_3. One may denote that S_2 is the longest common subsequence (LCS or LCSS) between S_1 and S_3 among all the possible subsequence. Indeed, the sequence [AC], [GT] or all other subsequences of S_2 are, in a transitive way, subsequences of S_1 and S_3. In other words, the longest common subsequence between two given strings must be a subsequence of both, and no other subsequence should be greater than it. In the previous example the longest common subsequence S_2 can be denoted as follows:

$$LCSS(S_1, S_3) = S_2$$

2.2.2 LCSS-BASED METHODS

Templates matching methods (TMMs) (Hartmann & Link, 2010) based on Dynamic Time Warping (Hartmann & Link, 2010), were demonstrated as non-efficient in presence of noisy raw signals (Vlachos *et al.*, 2003). To handle such data, Long-Van *et al.* (2012) have introduced two new methods, based on Longest Common Subsequence (LCSS), SegmentedLCSS and WarpingLCSS. Both SegmentedLCSS and WLCSS share the same training phase. This training allows converting accelerometer data into strings. This is due to the fact that LCSS is based on a problem that relies upon strings. In this way, raw signals must be quantized. The quantization step, proposed in (Long-Van *et al.*, 2012), involves computing clusters upon the training data with the K-Means algorithm. The resulting cluster centroids are

associated with pre-defined symbols to form strings. Therefore, each gesture instance is represented as a sequence of symbols. A LCSS score is associated with each sequence. The higher the LCSS score is between two elements, the greater is the similarity. Thus, a gesture instance is defined as a temporary template. The final motif is chosen based on the one with the highest average LCSS score. However, in order to be able to compute whether a signal belongs to a gesture class or not, a rejection threshold is associated with the template. This threshold is defined as the minimum LCSS between the previously elected template and all other gesture instances of the same class. Yet, L. V. Nguyen-Dinh, A. Calatroni and G. Tröster (2014) have suggested a new rejection threshold calculation, based on the mean μ_c and standard deviation σ_c of LCSS scores for the given class c. The resulting threshold ε is defined as $\varepsilon = \mu_c - h \cdot \sigma_c$, where h is an integer that allows adjusting the sensitivity of the algorithm for this class.

In the Segmented LCSS recognition process, the stream is stored in a sliding window *OW*. Each sample of this window is associated with previously generated centroids and its related symbol, based on the minimum Euclidean distance. Then, this new string is entirely compared to the template computed during training. If the resulting score exceeds the rejection threshold, of the associated class, then the gesture is associated with *c*. However, a gesture may be spotted as belonging to more than one class. To resolve such conflicts, a resolver may be added, as proposed in (Long-Van *et al.*, 2012). It is based on the normalized similarity *NormSim(A, B) = LCSS(A,B)/max(‖A‖, ‖B‖)*, where ‖A‖ and ‖B‖ are respectively

the length of A and B strings. The class with the highest NormSim is then marked as recognized. However, the SegmentedLCSS method implies to recompute the score each time the sliding window is shifted. As a result, the computation time is $O(T^2)$ (with T the size of the longest template) in the worst case. However, without OW the LCSS algorithm cannot find boundaries of incoming gestures. In this way, Long-Van *et al.* (2012) have introduced a new variant of the LCSS called Warping LCSS (WLCSS).

The WLCSS method removes need of a sliding window and improves the computational cost as it automatically determines gesture boundaries. In this new variant, quantized signals are still compared to the template of a given class. Nevertheless, this version only updates the score for each new element, starting from zero. This score grows when a match occurs and decreases thanks to penalties otherwise. The penalty consists of a weighted Euclidean distance between symbols, whether it is a mismatch, a repetition in the stream or even in the template. In a newer version presented in (L. V. Nguyen-Dinh *et al.*, 2014), the distance is normalized. Once the matching score is updated, the final result is output by the same decision maker used in the SegmentedLCSS method. The resulting time complexity for this new method is $O(T)$. Although the computational cost WLCSS is one order of magnitude lower than the SegmentedLCSS, the memory usage remains $O(T^2)$ in the worst case.

Recently, Roggen *et al.* (2015) have proposed a new, microcontroller optimized, version of the WLCSS algorithm called Limited Memory and WLCSS (LM-WLCSS). Identically to previous methods, this one is designed to spot motif in noisy raw signals and focuses on a single sensor channel. In this way, a quantization step may not be required. Moreover, the training phase of this new variant has also been modified in order to be embedded. This new step consists of recording all gestures, and defining the first instance as the template. The rejection threshold for this template is then computed thanks to the LM-WLCSS instead of the LCSS. As the WLCSS has edged issues, authors have modified the formula, and the resulting matching score is computed as follows:

$$
M_{j,i} = \begin{cases} 0 & \text{, if } i \leq 0 \text{ or } j \leq 0 \\ M_{j-1,i-1} + R & \text{, if } |S_i\text{-}T_j| \leq \varepsilon \\ \max \begin{cases} M_{j-1,i-1} - P \cdot (S_i\text{-}T_j) \\ M_{j-1,i} - P \cdot (S_i\text{-}T_j) \\ M_{j,i-1} - P \cdot (S_i\text{-}T_j) \end{cases} & \text{, if } |S_i\text{-}T_j| > \varepsilon \end{cases}
$$

Where S_i and T_j are respectively defined as the first i sample of the stream and the first j sample of the template. The resulting score, $M_{j,i}$, start from zero and increases of reward R, instead of just one, when the distance between the sample and the template does not exceed a tolerance threshold ε. Otherwise, the warping occurs and the matching score $M_{j,i}$ decreases of a penalty different from the WLCSS. This last one is always equal to the weighted distance between S_i and T_j, instead of relying on a mismatch, that is to say, a repetition in the stream or even in the template. Then, the resulting updated score is given to a local maximum searching algorithm called SearchMax, which filters scores exceeding the threshold within a sliding window of size W_f. Then, a one-bit event is sent whether a gesture is spot or not. When a match

occurs, the start point of the gesture may be retrieved by backtracking signals. This is performed *via* a window of size W_b to reduce unnecessary stored elements. Thus, the overall memory usage, for a word of size ws, is defined by $NT \times ws + NT \times W_b$ with NT representing the size of the template.

Moreover, in order to be able to manage multiple acquisition channels with the LM-WLCSS technique, two fusion methods were proposed. They are: the signal fusion (Long-Van *et al.*, 2012; L. V. Nguyen-Dinh *et al.*, 2014) and the decision fusion (Bahrepour, Meratnia, & Havinga, 2009; Zappi, Roggen, Farella, Tröster, & Benini, 2012). Observed performance evaluations with these usages were obtained from the Opportunity "Drill run", representing 17 distinct activities, and from 1 to 13 nodes. The resulting FScore is 85% for the decision fusion and 80% for the signal one. It demonstrates that higher is the number of nodes, better is the recognition performance.

2.3   CHAPTER CONCLUSION

The ending chapter was a small introduction to two techniques that may be employed as a basis for gesture recognition systems and a review on some methods relying on them. In this thesis, we choose to extend the LM-WLCSS algorithm as it is promising and it defines an improvement of the last version introduced by the same authors. Even though other methods relying on the LCSS have been proposed by Chen and Shen (2014), there is no previous work, to the best of our knowledge, that focus on a class optimization of the LM-WLCSS and perform a final decision fusion

with another classifier. Hence, we introduce in this work a new variant of the LM-WLCSS that preserves the capability to handle multi-class, as well as, a straightforward optimization for the quantization and the windows size W_f.

CHAPTER 3

A NEW OPTIMIZED LIMITED MEMORY AND WARPING LCSS

In this section, we introduce the Optimized LM-WLCSS (OLM-WLCSS), our proposed approach for online gesture recognition. This technique is robust against noisy signals and strong variability in gesture execution as well as methods we previously described. This section first describes the quantization step, following in the training phase. Then, the recognition block for one class and the optimization process are presented. Finally, we describe the decision-making module.

3.1   QUANTIZATION

Similarly to the WLCSS, we use K-Means algorithm to cluster the N_c data of the sensor in the quantization step. Each sample from the sensor is represented as a vector (e.g. an accelerometer is represented as a 3D vector). Thus, each sensor vectors are associated with their closest cluster centroid by comparing their Euclidean distances. Since the WLCSS does store symbols (as a representation of centroids), we suggest preserving centroids instead.

3.2   TRAINING

This subsection presents the overall vision of our offline training method in one class c. In the case of two or more classes, the process is repeated. Templates matching methods find similarities in the signal and detect gesture *via* a motif. The template can be elected as the best representation over the whole possible alternatives of the gesture in a training phase. Such patterns maximize the recognition performance. The overall process of our training is illustrated in Figure 12. Raw signals are first quantized to create a transformed training set. Next, this new data set is used for electing a template. Finally, resulting motif is given, as a parameter, to the rejection threshold calculation method that output the tuple (template, threshold).
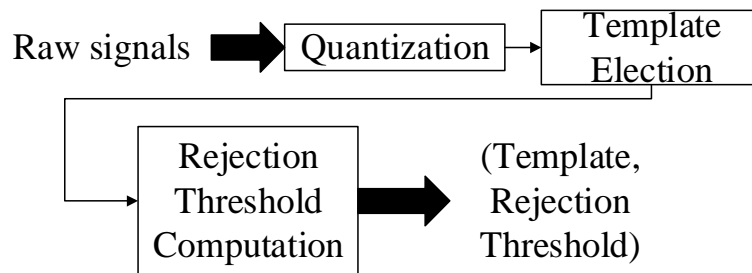
Figure 12: Overall training flow.

3.2.1 TEMPLATE ELECTION

Once the quantization phase is achieved, the next step is to elect the best template. As described in (Long-Van *et al.*, 2012), such process is performed *via* the LCSS method that has been modified to handle vector instead of symbols. Each instance was defined as a temporary template and then compared to the other ones. The reference template is defined thanks to the mean resulting score.

3.2.2 OLM-WLCSS

The core component of the presented method is the computation of the matching score. This is achieved thanks to the following formula:

$$M_{j,i} = \begin{cases} 0 & \text{if } i \leq 0 \text{ or } j \leq 0 \\ M_{j-1,i-1} + R & \text{if } d(S_i,T_j) = 0 \\ \max \begin{cases} M_{j-1,i-1} - P \\ M_{j-1,i} - P \\ M_{j,i-1} - P \end{cases} & \text{if } d(S_i,T_j) \neq 0 \end{cases}$$

Equation 5: Matching score equation

$$P = \beta \cdot d(S_i,T_j)$$

Equation 6: Penalty equation

Let d be the Euclidean distance between two centroids, S_i the i-th value of the quantized stream, and T_j the j-th value of the template. Identically to its predecessors, the initial value of the matching score $M_{j,i}$ is zero. Then, this score is increased by the value of R for every match when S_i equal to T_j. Otherwise, a penalty P weighted by β is applied. The resulting penalty is expressed according to three distinct cases. Firstly, when a mismatch between the stream and the template occurs. Secondly, when there is a repetition in the stream and finally, when there is a repetition in the template. Similarly to the LM-WLCSS, only the last column of the matching score is required to compute the new one. It should be noted that a backtracking method can be implemented to retrieve the starting point of the gesture.

3.2.3 REJECTION THESHOLD CALCULATION

The rejection threshold calculation is similar to the one presented in the LM-WLCSS algorithm. The score between the template and all the gesture instances of class c is computed with the core component of our algorithm. Then, the matching score mean μ_c and the standard deviation σ_c are calculated. The resulting threshold is determined by the following formula:

$$\text{Thd} = \mu - h \cdot \sigma, \quad h \in \mathbb{N}$$

Equation 7: Threshold equation

3.3 RECOGNITION BLOCKS FOR ONE CLASS

The outcome of the previous phase is the best tuple (template, rejection threshold) for each class. These two elements define parameters that allow matching a gesture to the incoming stream. Figure 13 illustrates the recognition flow. As for the training, raw signals are first quantized. The resulting sample and the previously elected template are given to the OLM-WLCSS method presented in the training phase. Next, the matching score is given to the SearchMax algorithm that sends a binary event.
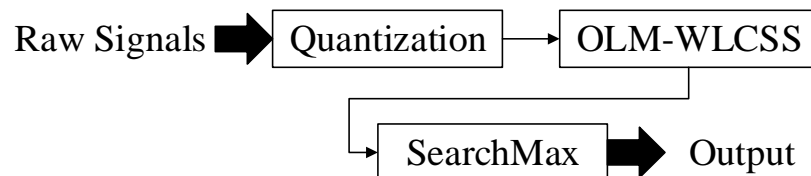
Raw Signals ➡ Quantization → OLM-WLCSS → SearchMax ➡ Output

Figure 13: Overall single class recognition flow.

3.3.1 SEARCHMAX

The matching score computed in previous steps should increase and exceed the threshold if a gesture is performed. However, noisy signals imply fluctuations and undesired detections. To overcome such issues, we used the SearchMax algorithm which was introduced in (Roggen *et al.*, 2015). Its goal is to find local maxima among matching scores in sliding window W_f. SearchMax loops over the scores and compares the last and the current score to set a flag; 1 for a new local maximum (Max_{sm}) and 0 for a lower value. A counter (K_{sm}) is increased at each loop. When K_{sm} exceeds the size of W_f the value of Max_{sm} is compared to the threshold *Thd*. Eventually, the algorithm returns a binary result; 1 if the local maximum is above *Thd* to indicate that a gesture has been recognized, 0 otherwise.

3.4 QUANTIZATION AND SEARCHMAX OPTIMIZATION

The previously described quantization phase associates each new sample to the nearest centroid of the class *c*. Thus, each class has a parameter k_c that defined the number of clusters generated in the training phase. In prior work, Long-Van *et al.* (2012) have defined it with a value of 20 after they ran some tests. In this way, we have also performed some tests with various cluster numbers. It appears that this parameter highly impacts the performance of the algorithm. Thus, we propose a straightforward optimization as illustrated in Figure 14. This step consists of

iteratively running the training process with different k_c. Therefore, we define $[2,$ $\sqrt{N_c}]$ as boundaries for k_c, where N_c is the number of samples used for the training of the class c. For the same reason, we tried to vary the sliding windows W_f we previously introduced, and noticed better performances from one to another. Consequently, we choose to adopt the same way as for k_c, and increment W_f from zero to twice the template size. The resulting best pair is elected based on its performance. To perform the evaluation, we decide to base the vote on the Cohen Kappa, as advice by Ben-David (2007), instead of accuracy that could be high due to a mere chance. The Kappa is computed from observed probabilities (P_o) and expected ones (P_e) as follows:

$$Kappa = \frac{P_o - P_e}{1 - P_e}$$

Equation 8: Kappa equation

38
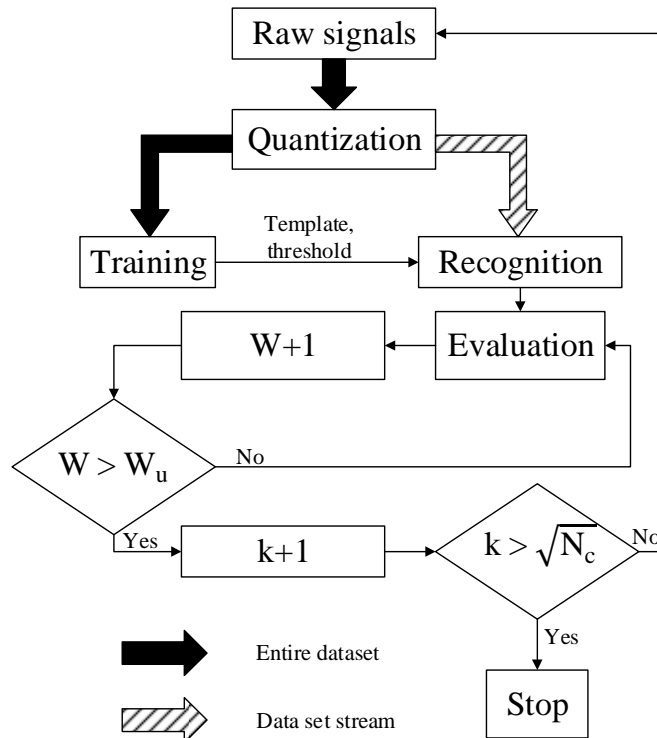
3.5 FINAL DECISION

Previous steps were independently performed for each gesture class. However, noise in raw signals and high variations in gesture execution can lead to multiple detections. Several methods are available to resolve conflicts, such as the weighted decision described in (Banos, Damas, Pomares, & Rojas, 2012). In our system, we choose to employ the lightweight classifier C4.5 (Quinlan, 2014), that requires a supervised training. The overall representation of the recognition flow is illustrated in Figure 15.

The training of C4.5 comes directly after the optimization step. It is performed using a 10-Fold cross-validation on a data set previously created. This file may be considered as a $N * M$ matrix, with N is the number of samples from the template training data set, and M is the amount of recognition blocks. Each element $r_{i,j}$ of this matrix represents the result of the j-th recognition block for the i-th sample.
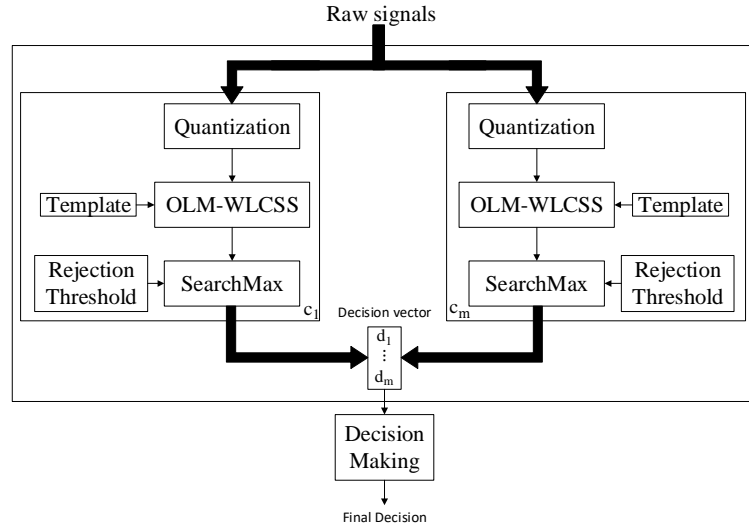
Figure 15: Overall recognition flow for m class.

3.6 DATA USED FOR OUR EXPERIMENTS

In order to evaluate the reliability of our algorithm, we have exploited two different data sets. None of these sets are the ones used in (Roggen *et al.*, 2015). Indeed, in (Roggen *et al.*, 2015) results were obtained on a private data set with arbitrary parameter. In this way a proper comparison with this algorithm is not possible.

Table 2: All gestures of Make Coffee data set

Make Coffee Gestures		
opening the brew basket lid (G1)	getting the measuring spoon (G6)	getting the decanter (G11)
pushing the shower	adding six spoons of	pouring the water

head (G2)	coffee in the filter (G7)	into the water reservoir for 5 seconds (G12)
putting filter into the filter basket (G3)	putting away the measuring spoon (G8)	putting the decanter onto the warmer plate (G13)
putting the coffee box in front of the coffeemaker (G4)	closing the coffee box (G9)	and closing the water lid (G14)
opening the coffee box (G5)	putting away the coffee box (G10)	

The first focuses on a unique activity, *to make coffee*. This activity is repeated 30 times. Since such an activity admits 14 distinct gestures, we have split them into 14 classes as enumerated in Table 2. The data set was created from data that came from two 9-DoF inertial measurement units (LSM9DS0). Each sensor was associated with an Intel Edison platform which was powered by a Lithium battery. The sampling rate of IMU was fixed at 20 Hz as advice by Karantonis, Narayanan, Mathie, Lovell and Celler (2006), indeed, most of body movements are largely under such a frequency. Once the configuration of IMUs was completed, the two nodes were placed on the subject's wrists. Data were sent to the computer *via* Wi-Fi. To record the activity, two members of our team have been selected. The first one was making coffee inside our laboratory, while the other one was labeling each incoming sample.

To ensure a good execution, the activity was achieved several times by the subject, as training, without any recording.

The second data set we use was suggested by the Bilkent University (Altun, Barshan, & Tunçel, 2010). It includes data from eight subjects, where each of them wore five 9-DoF inertial measurement units (IMU). The data set represents 19 daily or sports activities enumerated in Table 3. The realized experiment only exploits records from the first subject.

Table 3: All gestures of Bilkent University data set

Bilkent University Gestures		
Sitting (A1)	moving around in an elevator (A8)	exercising on a cross-trainer (A14)
Standing (A2)	walking in a parking lot (A9)	cycling on an exercise bike in horizontal and vertical positions (A15-16)
lying on back and on right side (A3-4)	walking on a treadmill with a speed of 4 km/h (in flat and 15 deg inclined positions) (A10-11)	rowing (A17)
ascending and descending stairs (A5-6)	running on a treadmill with a speed of	jumping (A18)

	8 km/h (A12)	
standing in an elevator still (A7)	exercising on a stepper (A13)	playing basketball (A19)

3.6.1 EVALUATION METRICS

The performance of the presented method was evaluated on three well-known metrics: Accuracy (Acc), FScore and Kappa measures. However, the last one was prioritized and provides the recognition performance of our algorithm. The first two were included as comparison purpose since they are widely used in classification problems. The FScore is based on the precision expressed by, $precision = \frac{TP}{TP + FP}$ and the recall $recall = \frac{TP}{TP + FN}$. Where TP is true positive values, FP false positives, TN true negatives and FN false negatives. These values were obtained after computing a confusion matrix. The final overall formula for the FScore computation is given as follows:

$$FScore = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Equation 9: FScore equation

3.7 RESULTS AND DISCUSSION

This section presents and discusses results we obtain with the two previously described data sets. Figure 16 and Figure 17 summarize metric values for the data set Make Coffee on the training and testing sets respectively. Abscess values are the axis

taken into account for each iteration of the given method. We have taken different sensors into account for each run. 3 axes represent the accelerometer, 6 refer to the accelerometer and the gyroscope, 9 all the IMU and 18 the two IMUs. The ordinate represents the Kappa, FScore and Accuracy, expressed in percentages (%), for each combination.

Performance results on the Make Coffee data set shows a considerable drop in the Kappa measure between the training set and the testing set for every axis. The second data set presents similar result with a Kappa of 81% for the training set and 37% with the testing set.
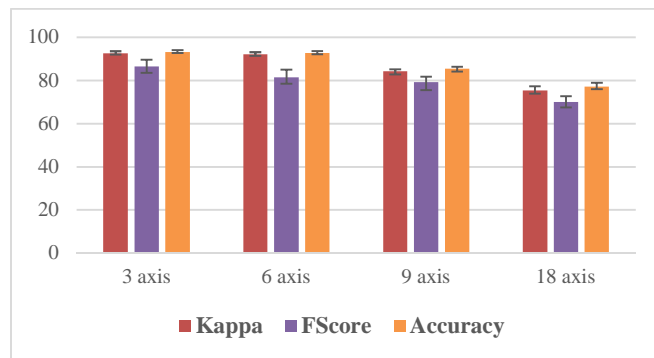
Figure 16: Results observed for the 10-Fold on the training set, for the make coffee data set.
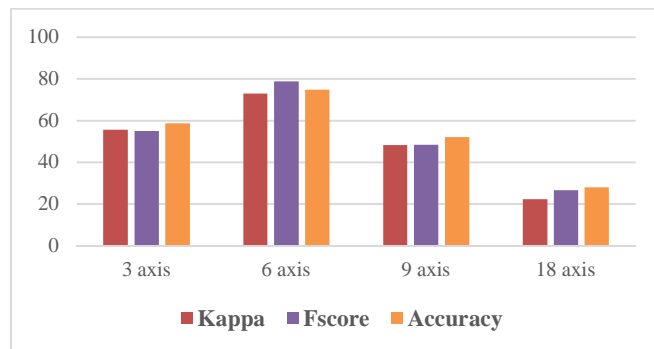
Figure 17: Results observed for supplied test set on the make coffee data set.

44

The observed difference between results we obtained, illustrates a significant limitation regarding the performance. This contrast may be due to both the optimization of parameters (such as clusters from K-Means and the size of the window for the SearchMax algorithm) and of each classifier over training data. We review some other method that falls in the same situation, good result on training set but low ones on testing set, that were identified as *overlearned* (Gamage, Kuang, Akmeliawati, & Demidenko, 2011). Indeed, as described by Witten and Frank (2005), a classifier trained and optimized on the same set will achieve accurate results on this one, but should fall down with independent test data. Consequently, our proposed method may be found in an *overlearning* situation, explaining such results. The cause is probably the fact that our method has parameters, as they must be optimized to achieve good results on test set. However, an optimization process will always constraint an algorithm to the optimization set.

3.8   CHAPTER CONCLUSION

In this chapter, we have proposed a new TMM derived from the LM-WLCSS technique, which aims at recognizing motifs in noisy streams. Several parameters were evaluated such as a suitable number of clusters for the quantization step, as well as, an adequate size of the window. The evaluation we have performed suggests promising results over the training set (92.7% of Kappa for 3-axis), but we have observed a serious drop with testing data (55.7% of Kappa for 3-axis). Such a contrast may be due to the fact that our method is overly dependent on the training data, which refers to the proper definition of an *overlearning* situation.

CHAPTER 4

GENERAL CONCLUSION

In the last decade with the emergence of MEMS, the literature on gesture recognition based on such devices has considerably grown. This has motivated the proposed master thesis project. The main purpose was to improve online gesture recognition systems. In the second chapter we were able to demonstrate the importance of gesture recognition systems and more precisely online gesture recognition systems. Moreover, we also demonstrate the importance of the accelerometer in such systems.

In this document we also reviewed multiple technique for gesture recognition based on accelerometer data, we starting with method based on the well-known distance measure DTW. However, these methods tend to be slow, that does not fit with the big data challenge. In this way we introduced methods relying on the LCSS problem that is modified to handle accelerometer data in a streaming way. More specifically we study the method of Roggen *et al.* (2015) and developed a new method from it that try to be more efficient and optimized. In addition, we tested our new method on some sets of more complicated gestures. However, we suspect the new model to fall in a "*overtraining*" state. Moreover, the recognition performance does not increase in comparison to previous work. This project was directed under a strict methodology that will be reviewed in the next section.

4.1   REALIZATION OF THE OBJECTIVES

In our methodology the first objective was to gain knowledge about the field that surrounds the problematic introduce in this master thesis. To realize this a review about important gesture recognition systems and technique was performed in the first place, more especially on the DTW distance measure that is extremely popular in the domain. Starting with general comprehension and utilization of this measure (Berndt & Clifford, 1994; Müller, 2007), and continuing with the one that employs accelerometer as the main sensors (Akl *et al.*, 2011; Akl & Valaee, 2010; Choe *et al.*, 2010). However, problems with this technique on accelerometer data were raised and another technique was studied: the LCSS (Cormen, Rivest, & Stein, 2009). More especially we review the series of methods relying upon the LCSS problem and introduced by (Long-Van *et al.*, 2012; L.-V. Nguyen-Dinh, A. Calatroni, & G. Tröster, 2014; L. V. Nguyen-Dinh *et al.*, 2014; Roggen *et al.*, 2015). These reviews led us to the contribution of this master thesis project.

The second phase enunciated in our methodology was to extend an existing online gesture recognition system in order to solve issues explained in the introduction document. The gesture recognition systems retained from our literature review is the LM-WLCSS presented by Roggen *et al.* (2015). Among all models explored in this review we find out it was one of the best and more particularly was introduced as a microcontroller optimized method with low-memory costs. Moreover, this model was easy to extend and was at the base of our new theoretical definition of

the new methods presented in this master thesis. This new method allows us to perform gesture recognition systems in a streaming way with complexes gesture from an IMU and answer issues raised in the introduction.

The third objective was to implement our new theoretical definition of the recognition systems to test it with real-world data. In this way a new software was developed in the Microsoft® oriented-object programming language C#. Moreover, another software in the programming language C++ was developed in order to get data from the IMU of the Intel® Edison development board. This resulting in exploiting raw data from the IMU in our new method that was charged with recognizing learned gestures.

The last objective of this master thesis project was to evaluate and validate our newly implemented theoretical definition of our new online gesture recognition systems. More precisely, the purpose was to validate the recognition rate of our new method with real-world data to evaluate how well it handles such data. In this way we recorded a new data set of us making coffee and break the activity into a subset of gesture for a total of 14 gestures. This scenario was repeated 30 times to have enough data for training and testing our method on separate data sets. Moreover, another well-known data set of physical activities was employed in our evaluation process. The results were then analyzed to draw conclusion on this objective.

4.2   PERSONNAL ASSESSMENT

In a conclusion of this master thesis project I would like to briefly dress a personal assessment of my first real experience in the world of research. I would say that this project was not the easiest part of my life and that it requires a solid motivation all the time. But to manage to successfully complete it I had to gain knowledge on my subject, gesture recognition, and it was really interesting. Moreover, as a non-native English speaker I had to acquire better reading and writing skills as it is the main language of the world of research. And more important I learn to build a strong methodology to success my project. This master thesis was also subject to produce a scientific publication that was unfortunately refused. As the last words I would say that we learned from our mistake, and I am thankful I was able to do a master thesis to acquire the necessary knowledge to pursue toward doctoral studies as I always wanted it.

REFERENCES

Akl, A., Feng, C., & Valaee, S. (2011). A Novel Accelerometer-Based Gesture Recognition System. *IEEE Transactions on Signal Processing*, *59*(12), 6197-6205. doi: 10.1109/TSP.2011.2165707

Akl, A., & Valaee, S. (2010, 14-19 March 2010). *Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing.* Presented at Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on,

Altun, K., Barshan, B., & Tunçel, O. (2010). Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognition*, *43*(10), 3605-3620. doi: 10.1016/j.patcog.2010.04.019

Bahrepour, M., Meratnia, N., & Havinga, P. J. M. (2009, 13-16 July 2009). *Sensor fusion-based event detection in Wireless Sensor Networks.* Presented at Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous '09. 6th Annual International,

Banos, O., Damas, M., Pomares, H., & Rojas, I. (2012). On the use of sensor fusion to reduce the impact of rotational and additive noise in human activity recognition. *Sensors*, *12*(6), 8039-8054.

Ben-David, A. (2007). A lot of randomness is hiding in accuracy. *Engineering Applications of Artificial Intelligence*, *20*(7), 875-885. doi: 10.1016/j.engappai.2007.01.001

Berndt, D. J., & Clifford, J. (1994). *Using Dynamic Time Warping to Find Patterns in Time Series.* Presented at KDD workshop,

Burgoon, J. K., Guerrero, L. K., & Floyd, K. (2016). *Nonverbal communication.* Routledge.

Chaudhary, A., Raheja, J. L., Das, K., & Raheja, S. (2013). Intelligent approaches to interact with machines using hand gesture recognition in natural way: a survey. *arXiv preprint arXiv:1303.2292.*

Chen, C., & Shen, H. (2014). Improving Online Gesture Recognition with WarpingLCSS by Multi-Sensor Fusion. In W. E. Wong, & T. Zhu (Éds.), *Computer Engineering and Networking* (Vol. 277, pp. 559-565): Springer International Publishing. doi: 10.1007/978-3-319-01766-2_64

Choe, B., Min, J.-K., & Cho, S.-B. (2010). Online Gesture Recognition for User Interface on Accelerometer Built-in Mobile Phones. In K. W. Wong, B. S. U. Mendis, & A. Bouzerdoum (Éds.), *Neural Information Processing. Models and Applications: 17th International Conference, ICONIP 2010, Sydney, Australia, November 22-25, 2010, Proceedings, Part II* (pp. 650-657). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-17534-3_80

Cormen, T. H. L., Charles E, Rivest, R. L., & Stein, C. (2009). Longest common subsequence. In *Introduction to algorithms* (pp. 390-397): MIT press.

Dardas, N. H., & Georganas, N. D. (2011). Real-Time Hand Gesture Detection and Recognition Using Bag-of-Features and Support Vector Machine Techniques. *Instrumentation and Measurement, IEEE Transactions on*, *60*(11), 3592-3607. doi: 10.1109/TIM.2011.2161140

Dardas, N. H. A.-Q. (2012). *Real-time hand gesture detection and recognition for human computer interaction*. Université d'Ottawa/University of Ottawa.

Frey, B. J., & Dueck, D. (2007). Clustering by Passing Messages Between Data Points. *Science*, *315*(5814), 972-976. doi: 10.1126/science.1136800

Gamage, N., Kuang, Y. C., Akmeliawati, R., & Demidenko, S. (2011). Gaussian Process Dynamical Models for hand gesture interpretation in Sign Language. *Pattern Recognition Letters*, *32*(15), 2009-2014. doi: http://dx.doi.org/10.1016/j.patrec.2011.08.015

Guiry, J. J., van de Ven, P., & Nelson, J. (2014). Multi-sensor fusion for enhanced contextual awareness of everyday activities with ubiquitous devices. *Sensors*, *14*(3), 5687-5701.

Hartigan, J. A. (1975). *Clustering algorithms*.

Hartmann, B., & Link, N. (2010, 10-13 Oct. 2010). *Gesture recognition with inertial sensors and optimized DTW prototypes*. Presented at Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on,

Hirschberg, D. S. (1977). Algorithms for the longest common subsequence problem. *Journal of the ACM (JACM)*, *24*(4), 664-675.

Hyeon-Kyu, L., & Kim, J. H. (1999). An HMM-based threshold model approach for gesture recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, *21*(10), 961-973. doi: 10.1109/34.799904

Ibraheem, N. A., & Khan, R. Z. (2012). Survey on various gesture recognition technologies and techniques. *International journal of computer applications*, *50*(7).

Jang, M., Han, M.-S., Kim, J.-h., & Yang, H.-S. (2011). Dynamic Time Warping-Based K-Means Clustering for Accelerometer-Based Handwriting Recognition. In K. G. Mehrotra, C. Mohan, J. C. Oh, P. K. Varshney, & M. Ali (Éds.), *Developing Concepts in Applied Intelligence* (pp. 21-26). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-642-21332-8_3

Karantonis, D. M., Narayanan, M. R., Mathie, M., Lovell, N. H., & Celler, B. G. (2006). Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring. *IEEE Transactions on Information Technology in Biomedicine*, *10*(1), 156-167. doi: 10.1109/TITB.2005.856864

Kılıboz, N. Ç., & Güdükbay, U. (2015). A hand gesture recognition technique for human–computer interaction. *Journal of Visual Communication and Image Representation*, *28*, 97-104. doi: 10.1016/j.jvcir.2015.01.015

Kim, J. H., Thang, N. D., & Kim, T. S. (2009, 5-8 July 2009). *3-D hand motion tracking and gesture recognition using a data glove*. Presented at 2009 IEEE International Symposium on Industrial Electronics,

Liu, J., Zhong, L., Wickramasuriya, J., & Vasudevan, V. (2009). uWave: Accelerometer-based personalized gesture recognition and its applications. *Pervasive and Mobile Computing*, *5*(6), 657-675. doi: 10.1016/j.pmcj.2009.07.007

Liu, Y., Yin, Y., & Zhang, S. (2012, 26-27 Aug. 2012). *Hand Gesture Recognition Based on HU Moments in Interaction of Virtual Reality*. Presented at Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on,

Long-Van, N.-D., Roggen, D., Calatroni, A., & Troster, G. (2012, 27-29 Nov. 2012). *Improving online gesture recognition with template matching methods in accelerometer data*. Presented at Intelligent Systems Design and Applications (ISDA), 2012 12th International Conference on,

Mitra, S., & Acharya, T. (2007). Gesture Recognition: A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, *37*(3), 311-324. doi: 10.1109/TSMCC.2007.893280

Müller, M. (2007). Dynamic Time Warping. In *Information Retrieval for Music and Motion* (pp. 69-84). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/978-3-540-74048-3_4

Nguyen-Dinh, L.-V., Calatroni, A., & Tröster, G. (2014). *Towards a unified system for multimodal activity spotting: challenges and a proposal*. Presented at

Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, Seattle, Washington.

Nguyen-Dinh, L. V., Calatroni, A., & Tröster, G. (2014). Robust online gesture recognition with crowdsourced annotations. *Journal of Machine Learning Research*, *15*, 3187-3220.

Pan, T. Y., Lo, L. Y., Yeh, C. W., Li, J. W., Liu, H. T., & Hu, M. C. (2016, 20-22 April 2016). *Real-Time Sign Language Recognition in Complex Background Scene Based on a Hierarchical Clustering Classification Method*. Presented at 2016 IEEE Second International Conference on Multimedia Big Data (BigMM),

Pylvänäinen, T. (2005). Accelerometer Based Gesture Recognition Using Continuous HMMs. In J. S. Marques, N. Pérez de la Blanca, & P. Pina (Éds.), *Pattern Recognition and Image Analysis: Second Iberian Conference, IbPRIA 2005, Estoril, Portugal, June 7-9, 2005, Proceedings, Part I* (pp. 639-646). Berlin, Heidelberg: Springer Berlin Heidelberg. doi: 10.1007/11492429_77

Quinlan, J. R. (2014). *C4. 5: programs for machine learning*. Elsevier.

Rautaray, S. S., & Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, *43*(1), 1-54. doi: 10.1007/s10462-012-9356-9

Reyes, M., Dominguez, G., & Escalera, S. (2011, 6-13 Nov. 2011). *Featureweighting in dynamic timewarping for gesture recognition in depth data*. Presented at Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on,

Rizwan, M., Rahmat-Samii, Y., & Ukkonen, L. (2015). *Circularly polarized textile antenna for 2.45 GHz*. Presented at RF and Wireless Technologies for Biomedical and Healthcare Applications (IMWS-BIO), 2015 IEEE MTT-S 2015 International Microwave Workshop Series on,

Roggen, D., Cuspinera, L., Pombo, G., Ali, F., & Nguyen-Dinh, L.-V. (2015). Limited-Memory Warping LCSS for Real-Time Low-Power Pattern Recognition in Wireless Nodes. In T. Abdelzaher, N. Pereira, & E. Tovar (Éds.), *Wireless Sensor Networks* (Vol. 8965, pp. 151-167): Springer International Publishing. doi: 10.1007/978-3-319-15582-1_10

Rung-Huei, L., & Ming, O. (1998, 14-16 Apr 1998). *A real-time continuous gesture recognition system for sign language*. Presented at Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on,

Schlömer, T., Poppinga, B., Henze, N., & Boll, S. (2008). *Gesture recognition with a Wii controller*. Presented at Proceedings of the 2nd international conference on Tangible and embedded interaction, Bonn, Germany.

Shoaib, M., Bosch, S., Incel, O., Scholten, H., & Havinga, P. (2015). A Survey of Online Activity Recognition Using Mobile Phones. *Sensors*, *15*(1), 2059.

Song, Y., Demirdjian, D., & Davis, R. (2012). Continuous body and hand gesture recognition for natural human-computer interaction. *ACM Trans. Interact. Intell. Syst.*, *2*(1), 1-28. doi: 10.1145/2133366.2133371

Thalmic Labs Inc. Myo. Retrieve from https://www.myo.com/

Vlachos, M., Hadjieleftheriou, M., Gunopulos, D., & Keogh, E. (2003). *Indexing multi-dimensional time-series with support for multiple distance measures*. Presented at Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, Washington, D.C.

Witten, I. H., & Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition (Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann Publishers Inc.

Zappi, P., Roggen, D., Farella, E., Tröster, G., & Benini, L. (2012). Network-Level Power-Performance Trade-Off in Wearable Activity Recognition: A Dynamic Sensor Selection Approach. *ACM Trans. Embed. Comput. Syst.*, *11*(3), 1-30. doi: 10.1145/2345770.2345781