BLSTM-RNN based 3D Gesture Classification

Grégoire Lefebvre[1], Samuel Berlemont[1,2],
Franck Mamalet[1], and Christophe Garcia[2]

[1]Orange Labs, R&D, France
{firstname.surname}@orange.com
[2]LIRIS, UMR 5205 CNRS, INSA-Lyon, F-69621, France.
{firstname.surname}@liris.cnrs.fr

Abstract. This paper presents a new robust method for inertial MEM (MicroElectroMechanical systems) 3D gesture recognition. The linear acceleration and the angular velocity, respectively provided by the accelerometer and the gyrometer, are sampled in time resulting in 6D values at each time step which are used as inputs for the gesture recognition system. We propose to build a system based on Bidirectional Long Short-Term Memory Recurrent Neural Networks (BLSTM-RNN) for gesture classification from raw MEM data. We also compare this system to a geometric approach using DTW (Dynamic Time Warping) and a statistical method based on HMM (Hidden Markov Model) from filtered and denoised MEM data. Experimental results on 22 individuals producing 14 gestures in the air show that the proposed approach outperforms classical classification methods with a classification mean rate of 95.57% and a standard deviation of 0.50 for 616 test gestures. Furthermore, these experiments underline that combining accelerometer and gyrometer information gives better results that using a single inertial description.

Keywords: LSTM-RNN, DTW, HMM, MEM, hand gesture recognition

1 Introduction

Accelerometers and gyrometers are nowadays present in our everyday Smartphones. These sensors capture hand movements when users grasp their devices. We can consider two main issues: posture recognition and symbolic gesture recognition. In the first case, the user maintains a posture during a certain period of time, describing for instance the fact that the device is upside down. In the second situation, the user may produce a gesture to execute a system command, like drawing a *heart* symbol in 3D space to call its favorite phone number. Dynamic gesture recognition based on inertial sensors is a very challenging task. Algorithms are confronted to numerous factors causing errors in the recognition process: dynamical differences (intensive versus phlegmatic gestures), temporal differences (slow versus fast movements), physical constraints (device weight, human body elasticity, left or right-handed, seated or standing up, on the move, etc.), classification constraints (mono versus multi users, open or closed world paradigm, etc.). Classically, several steps operate from signal data preprocessing

to gesture classification with some intermediate steps like data clustering and gesture model learning. The preprocessing steps aim at reducing the input signals that characterize the corresponding gestures. Different methods can then be applied: calibration, filtering, normalization or vectorization. Data clustering is often applied to reduce the input space dimension and find class referent gesture vectors. A learning phase of a gesture model follows this clustering step and finally a decision rule or a specific classifier is built to label the input data as a recognized gesture or an unknown gesture. In this article, we propose to learn an efficient gesture classifier without any preprocessing method (i.e. from raw MEM data) using a BLSTM-RNN model.

This paper is organized as follows. In Section 2, sensor-based gesture recognition is described with a survey. Section 3 presents our recognition method. Section 4 describes the experimental results. Finally, conclusions are drawn.

2 Accelerometer based 3D Gesture Recognition

3D gesture recognition using accelerometers has been studied in recent years, and for gesture classification three main strategies stand out which are based on statistics, on geometry or on boosting classifier approaches.

The first strategy has been deeply studied in the last decade with two main approaches: discrete versus continuous HMM [6–8, 11]. Hofmann et al. [6] proposed to use discrete HMM (dHMM) for recognizing dynamic gestures thanks to their velocity profile. This approach consists of two levels and stages of recognition: a low-level stage essentially dividing the input data space into different regions and assigning each of them (i.e. creation of a vector codebook), and a high-level stage taking the sequences of vector indexes from the first stage and classifying them with discrete HMM. The experiments are built using a training set with 10 samples per gesture, each sample representing hand orientation, acceleration data and finger joint angle. A vector codebook is obtained by an input space clustering method (i.e. K-means algorithm). Clustering essentially serves as an unsupervised learning procedure to model the shape of the feature vector distribution in the input data space. Here, the number of HMM states vary from 1 to 10 and the observation alphabet size equals to 120. The comparison between ergodic HMM and left-to-right HMM shows similar results with 95.6% correct recognition rate for 100 gestures. Similar results are presented in [7, 8]. Kallio et al. [7] use 5 HMM states and a codebook size of 8 for 16 gestures. The authors highlight that the performances decrease when using 4 sequences for training the system compared to 20 sequences. The recognition rate falls from 95% to 75% even for this mono-user case study. In [8], a 37 multi-user case is studied with 8 gestures, evaluating the effect of vector quantization and sampling. A rate of 96.1% of correct classification is obtained with 5 HMM states and a codebook size of 8. However, this study can be seen as biased since the K-means clustering is performed from all the available data set and not only the training database. In opposition to the previous studies, and to take into consideration that gesture data are correlated in time, Pylvänäinen proposes in [11] to build a system based

on continuous HMM (cHMM). Again, the results are convincing, with 96.76% on a dataset providing 20 samples for 10 gestures realized by 7 persons.

The second strategy for recognizing 3D gestures is based on geometric models with distance computation. The goal is to provide a gallery of some gesture references to model each gesture class and design a decision rule for a test gesture regarding the respective distance to these referent instances. On the contrary to the HMM strategy, no learning phase is needed but computational time is required for a test gesture to be compared to all referent instances. Consequently, the main drawback of this approach is the necessity to find the most relevant samples to represent a gesture class while keeping the number of these referents low in order to minimize the final evaluation processing time. Wilson et al. in [13] compare Linear Time Warping (LTW) and Dynamic Time Warping (DTW) to the HMM based strategy. Their experiment with 7 types of gesture from 6 users shows an advantage for HMM with 90% in opposition to the score of LTW and DTW of respectively 40% and 71%. Liu et al. experiment with more success the DTW strategy in [9]. Gesture recognition and user identification are performed with good recognition rates of respectively 93.5% and 88%. The authors introduce an averaging window of 50 ms for reducing noise and erratic moves. The gesture data, performed over multiple days, consists of 30 samples of 8 gestures for 8 individuals and the user recognition results are obtained from 25 participants. Likewise, in [2], Akl et al. use DTW and affinity propagation for dimension reduction for recognizing 3D gestures. 7 subjects participated producing 3700 gesture traces for a good classification rate of 90%.

The third strategy for recognizing 3D gestures is to learn a specific classifier. Hoffman et al. (see [5]) improve 3D gesture recognition with a linear classifier and Adaboost, inspired by the method proposed in [1] for 2D symbol writer recognition. The experiments show an accuracy of 98% for 13 gestures made by 17 participants. Other studies focus on SVM (i.e. Support Vector Machine) like in [14]. This study uses frame-based descriptors. Each gesture is divided into segments where are computed to form descriptors: mean, energy, entropy, standard deviation and correlation. These descriptors constitute the feature vector to be classified by a multi-class SVM. The obtained results are 95.21% of good recognition for 12 gestures made by 10 individuals.

Consequently, many strategies are explored with different paradigms and specific data processing methods on different databases. Nevertheless, theses approaches suffer from finding automatically the relevant parameters (e.g. signal processing, etc.) to deal with gesture variabilities. We develop hereafter our 3D gesture recognition method based on BLSTM-RNN from raw input data and compare it with classical methods on a common database.

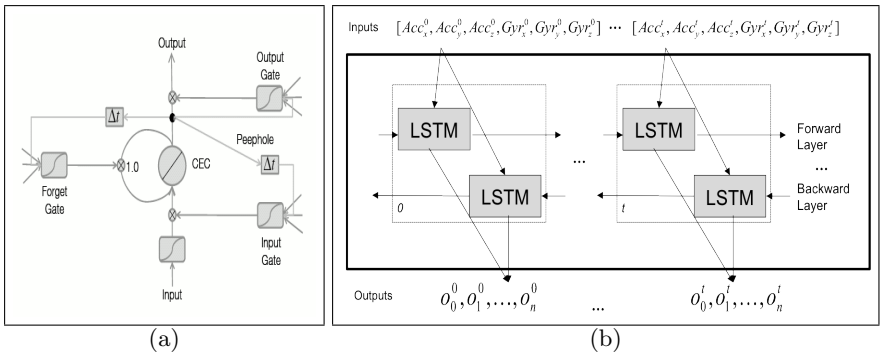
3 The proposed 3D Gesture Recognition Method

3.1 Bidirectional Long Short-Term Memory RNNs

Classical RNNs are a common learning technique for temporal analysis of data since they are able to take into consideration the temporal context. This is

achieved by using recurrent connections within the hidden layer which allow the network to *remember* a state representing the previous input values. However, Hochreiter and Schmidhuber in [12] have shown that if RNNs can handle short-time lags between inputs, the problem of *exponential error decay* prevent them from tackling real-life long-term dependencies. They introduced thus the Long Short Term Memory RNNs, that allows a constant error signal propagation through time using a special node called *constant error carousel* (CEC) and multiplicative gates (Fig 1.a). These gates are neurons that can set (input gate), reset (forget gate) or hide (output gate) the internal value of the CEC according to neuron input values and context.

LSTM-RNNs have proven their great ability to deal with temporal data in many applications (e.g. phoneme classification [4], action classification [3]). In this paper we consider gesture data using 6D input vectors through sampling timestep. These data are correlated during the user gestural production, and time lags between the beginning and the end of gesture can be long. For these reasons, LSTM-RNN is chosen to classify the input MEM data sequence. Furthermore, since gesture recognition, at a given timestep, may depend on past and future context, we use Bidirectional LSTM-RNN (BLSTM-RNN), introduced in [4], that consists in two separate hidden layers, the forward (resp. backward) layer able to deal with past (resp. future) context. The output layer is connected to both hidden layers in order to fuse past and future contexts.



(a) (b)

Fig. 1. (a) LSTM neuron. (b) BLSTM-RNN Architecture.

3.2 BLSTM-RNN Architecture, Training and Decision Rule

The proposed gesture classification scheme based on BLSTM-RNN is described in Figure 1.b. First, the input layer consists in the concatenation of accelerometer and gyrometer information synchronized in time (i.e. 6 input values per timestep). Notice that our system relies only on the raw MEMs data, without any preprocessing in opposition to most of state-of-the-art methods. These data

are linearly normalized between -1 and +1 according to the maximum value that sensors can provide. The forward and backward LSTM hidden layers are fully connected to the input layer and consist in 100 LSTM neurons each with full recurrent connections. The output layer has a size equals to the number of gesture to classify. The SoftMax activation function is used for this layer to give network responses between 0 and 1 at every timestep. Classically, these outputs can be considered as posterior probabilities of the input sequence to belong to a specific category at a given timestep. This network is learned using classical on-line backpropagation through time with momentum (i.e. learning rate $5e{-}4$, momentum 0.2), as described in [12], on a training set, by targeting the same corresponding gesture class at each time step for each input example. For evaluation of a new gesture sequence, we use a majority voting rule over the outputs along the sequence (i.e. keeping only the most *probable* class at each time step) to determine the final gesture class.

4 Experimental Results

There is no public dataset for comparison of 3D gesture recognition. Therefore, we have collected our 3D gesture dataset to compare classification methods. Our dataset has been captured on an Android Nexus S Samsung device. 22 participants, from 20 to 55 years old, all right-handed, performed 5 times each of the 14 symbolic gestures. This corresponds to 1540 temporal segmented gestures. The sampling time for accelerometer and gyroscope capture is 40 ms. The 14 symbolic gestures are divided into 2 families: linear gestures (e.g. *north*, *south*, *east* and *west flicks*, and *up*, *down*, *pick* and *throw* gestures) and curvilinear gestures (e.g. *alpha*, *heart*, *letter N*, *letter Z*, *clockwise* and *counter-clockwise*). These choices make the dataset difficult. There are classically confusions between *flick* gestures and *letter N* and *Z*. Likewise, the *clockwise* movement is often confused with *alpha* or *heart* symbols. Hereafter, we use temporal segmented gestures where only useful data are efficient to classify the inputs.

We use 3 different configurations to compare our solution based on BLSTM-RNN to 3 state-of-the-art solutions: DTW, dHMM and cHMM based methods. The DTW solution uses a 5 nearest neighbor classification [10] and the HMM solution uses the maximum of likelihood as a decision rule . In all experiments, we use a filtered and vectorized gestural information for these methods and raw MEM information for LSTM solution. In the following, we use a 3-fold cross validation.

The first configuration (DB1) corresponds to the personalization paradigm, where only one user is considered with few learning examples. For this configuration we have used the 70 gestures of a single participant in the learning phase, and ask him to process 16 more instances of each gesture for test (i.e. 224 gestures). The second configuration (DB2) uses 3 instances of each gesture per user for the learning phase: 924 gestures (i.e. 60% of all data) are used for the learning phase and 616 gestures (i.e. 40%) for the test phase. This case corresponds to a multi-user system and a closed world paradigm. The third configuration

(DB3) is composed of all samples from 17 users (i.e. 1190 gestures) and the test data uses the other available gestures (i.e. 350 gestures from unknown users). This case is close to a real system trained with a few examples and having to generalize to new users who want to use it without any personalization phase. Here, the configuration represents the open world paradigm.

Table 1. Good classification rates on DB1, DB2 and DB3.

Databases	DB1	DB2	DB3
Methods	Mean & Standard Deviation		
DTW acc	99.40% ± 0.21%	92.59% ± 0.20%	90.29% ± 2.07%
DTW gyro	95.39% ± 0.56%	80.63% ± 2.39%	79.81% ± 1.72%
DTW acc+gyro	99.70% ± 0.42%	94.04% ± 0.15%	91.71% ± 1.46%
dHMM acc	77.14% ± 5.18%	64.09% ± 1.60%	63.81% ± 0.58%
dHMM gyro	57.50% ± 3.24%	43.13% ± 2.35%	49.05% ± 1.15%
dHMM acc+gyro	81.02% ± 3.72%	69.46% ± 2.11%	66.95% ± 1.87%
cHMM acc	99.02% ± 0.81%	83.99% ± 1.09%	80.09% ± 2.82%
cHMM gyro	95.05% ± 2.62%	70.92% ± 0.74%	70.76% ± 0.58%
cHMM acc+gyro	**99.86% ± 0.02%**	85.79% ± 0.67%	82.76% ± 1.41%
BLSTM-RNN acc	84.15% ± 0.67%	94.86% ± 1.23%	89.42% ± 2.45%
BLSTM-RNN gyro	68.90% ± 4.85%	83.39% ± 0.65%	74.19% ± 1.55%
BLSTM-RNN acc+gyro	86.75% ± 0.75%	**95.57% ± 0.50%**	**92.57% ± 2.85%**

Classification Results Table 1 outlines the global performances of each classifier for configurations DB1, DB2 and DB3 coupling or not accelerometer and gyrometer data. Considering coupled input data (accelerometer+gyroscope), this table shows that our BLSTM-RNN based classifier gives the best results on DB2 and DB3, with respectively $95.57 \pm 0.50\%$ and $92.57 \pm 2.85\%$.

In the three configurations, the dHMM solution provides lower performances which is mainly due to the input data variability and the complexity to determine an automatic discriminant codebook.

On two configurations (DB2 and DB3), the DTW solution achieves the second best performance in mean recognition rate before the cHMM based one.

On DB1 configuration, DTW and cHMM achieve equivalent performances while our BLSTM-RNN approach is less efficient. This is mainly due to the lack of learning data which leads to the classical over-fitting issue. The attempts made with smaller LSTM networks did not allow any improvement on generalization.

When comparing these methods using a single input MEM sensor (accelerometer or gyroscope), we can see that using only gyroscope data is less efficient than using single accelerometer data. Moreover, when these two information are combined, the performances increase with respectively $99.70 \pm 0.42\%, 94.04 \pm 0.15\%$ and $91.71 \pm 1.46\%$, for instance, for the DTW based method on DB1, DB2 and DB3 configurations.

Main conclusions of a deep analysis of confusion matrices (not provided here due to lack of space) are the following. The main drawback for the cHMM based

method in this context is the incorrect classification of the N gestures with only 0.95% of correct classification. 62.86% of the N gestures are confused with the pick gestures. A strong confusion appears with opposite gestures as pick and throw or down and up gestures. Opposite gestures may be mis-classified when some user anticipate a north flick gesture by slightly moving back the device in the beginning of the production. On the contrary, the DTW based method provide a good solution to classify linear gestures except for the *throw* gesture which is often recognized as *east* and *north flicks*, which can be explained by the similar nature of production of these three gesture types. Our BLSTM-RNN approach have some issue to distinguish the *east flick* gesture from the *letter Z* and the *up* gesture from the *letter N*, both sharing the same initial movement. This may be due to the uniform learning target chosen (same class at each time step), or the majority voting scheme in recognition phase.

Table 2. Computing time (in ms) to classify one unknown gesture.

Databases	DB1	DB2	DB3
Leaning samples	70	924	1190
Test samples	224	616	350
DTW accgyro	11.93 ±0.02	34.57 ±0.47	44.58±0.38
dHMM accgyro	18.31 ±0.17	24.84 ±0.32	16.18±0.32
cHMM accgyro	42.53 ±1.97	23.89±2.74	30.19±1.65
BLSTM-RNN accgyro	30.47± 0.23	31.12±0.57	29.56±0.48

Computing Times Table 2 presents the computing times for all methods for the 3 configurations in recognition phase executed on an Intel Core i5 CPU at 2.67 GHz with 3.42 Go of RAM. These experimental results show that the computing time for the BLSTM-RNN and HMM based solutions is quite constant regarding the tasks on the different database (i.e. around 30 ms for BLSTM-RNN and 18 ms for dHMM to classify one input gesture for DB1). The learning process is built indeed off-line and consequently the recognition process is fast. On the contrary, the DTW solution requires to compare the input gesture with all learning reference samples. That is why the computing time increases in mean from 11.93 ms for 70 learning samples to 44.58 ms for 1190 learning samples. The DTW solution requires a small number of reference gestures and which makes it hard to cover all user gesture variations. Consequently, the proposed system, based on BLSTM-RNN, achieving the best result performances in multi-user configuration with a recognition computing time independent of training dataset size is a very challenging solution.

5 Conclusion and Perspectives

In this paper, we have presented a contribution based on BLSTM-RNN and a comparison for inertial MEM based gesture recognition. This study about sym-

bolic gesture recognition compares our contribution to 3 classical pattern recognition methods: the geometric approach using DTW and the statistical method based on dHMM and cHMM. We have shown that on multi-user configuration our approach achieves the best mean classification rates, up to 95.57%, in a closed world configuration. Main remaining confusions with the proposed solution are when two 3D trajectories are similar or share some initial movements, as an east flick and a Z letter. New approach using a modified objective function, such as Connectionist Temporal Classification [4], that permits to jointly learn to localize and classify events in input sequences, might be used to overcome this issue or to classify non segmented gestures.

References

1. A practical approach for writer-dependent symbol recognition using a writer-independent symbol recognizer. *IEEE Trans. PAMI*, 29(11):1917–1926, 2007.
2. A. Akl and S. Valaee. Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing. In *ICASSP*, 2010.
3. M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt. Sequential Deep Learning for Human Action Recognition. In *HBU*, Lecture Notes in Computer Science, pages 29–39, Nov. 2011.
4. A. Graves and J. Schmidhuber. Framewise phoneme classification with bidirectional lstm and other neural network architectures. *Neural Networks*, (18):5–6, 2005.
5. M. Hoffman, P. Varcholik, and J. LaViola. Breaking the status quo: Improving 3d gesture recognition with spatially convenient input devices. In *Virtual Reality Conference (VR)*, pages 59 –66, 2010.
6. F. Hofmann, P. Heyer, and G. Hommel. Velocity profile based recognition of dynamic gestures with discrete hidden markov models. In *Gesture and Sign Language in Human-Computer Interaction*, volume 1371 of *Lecture Notes in Computer Science*, pages 81–95. 1998.
7. S. Kallio, J. Kela, and J. Mantyjarvi. Online gesture recognition system for mobile interaction. In *Systems, Man and Cybernetics*, volume 3, pages 2070 – 2076 vol.3, 2003.
8. J. Kela, P. Korpipää, J. Mäntyjärvi, S. Kallio, G. Savino, L. Jozzo, and D. Marca. Accelerometer-based gesture control for a design environment. *Personal Ubiquitous Comput.*, 10(5):285–299, July 2006.
9. J. Liu, Z. Wang, L. Zhong, J. Wickramasuriya, and V. Vasudevan. uwave: Accelerometer-based personalized gesture recognition and its applications. In *IEEE PerCom*, pages 1 –9, 2009.
10. E. Petit. GRASP: Moteur de reconnaissance de gestes. Technical report, France Télécom R&D, 2007.
11. T. Pylvänäinen. Accelerometer Based Gesture Recognition Using Continuous HMMs Pattern Recognition and Image Analysis. volume 3522 of *Lecture Notes in Computer Science*, chapter 77, pages 413–430. Berlin, Heidelberg, 2005.
12. H. S. and J. Schmidhuber. Long short-term memory. *Neural computation*, (9):1735–1780, 1997.
13. D. H. Wilson and A. Wilson. Gesture recognition using the xwand. Technical Report CMU-RI-TR-04-57, Robotics Institute, April 2004.
14. J. Wu, G. Pan, D. Zhang, G. Qi, and S. Li. Gesture recognition with a 3-d accelerometer. UIC '09, pages 25–38, 2009.