Example of PhD Thesis with RoboticsLaTeX template

Barbara Bruno, Fulvio Mastrogiovanni

DIBRIS - Department of Computer Science, Bioengineering, Robotics and System Engineering

University of Genova

In partial fulfillment of the requirements for the degree of

Doctor of Philosophy

February 31, 2015

Acknowledgements

Don't forget to acknowledge your supervisor!

To all the Master and PhD students of Robotics Engineering at the University of Genova.

Abstract

This is a very short and uninformative abstract.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivations

HRI, gesture recognition, etc.

1.2 Context of the Study

Descizione possibile scenario

1.3 Objectives and Contributions

Problemi da risolvere e problemi risolti

1.4 Overview of the Thesis

Descrizione delle sezioni della tesi

Chapter 2

State of the Art

> In order for a program to be capable of learning something it must first be capable of being told it.
>
> *"Programs with Common Sense"*
> *John McCarthy, 1959*

Introduction

Human gesture recognition consists of identifying and interpreting purposive human motions using different types of sensors. In this chapter, an up-to-date state-of-the-art in human gesture recognition is presented, which includes includes gesture taxonomies, gesture modeling and recognition techniques. To study gesture recognition, it is necessary to understand the definition and the nature of gestures. If we try to define the word 'gesture' we figure out that there are many aspects to take into account, such as different categories of gestures, the reason why we use particular gestures and also what kind of information can be conveyed by them. All this aspects are treated in section 2.1. In section 2.2, two main categories of existing technologies for human gesture recognition are introduced according to the sensor type: vision-based approaches and approaches not considering visual information. Subsection 2.3 overviews of the main methods used by human gesture recognition approaches to represent and classify gestures. Section 2.4 briefly explains the proposed method and the main contributions of this research.

2.1 Taxonomy of Gestures

Gestures constitute an intuitive and natural means of human communication. They are used in order to convey information or to interact with the environment. Gestures can reveal intentions or attitudes, can correspond to signs or signals and also can be part of actions. Since there is no universal meaning for a gesture, the meaning of a gesture strongly differs among cultures [1], different categorization and different taxonomies of gestures can be defined with respect to different criteria.

In particular, gestures can be classified according to their function [2],[3]:

1. Semiotic: those used to communicate meaningful information
2. Ergotic: those used to manipulate the physical world and create artifacts
3. Epistemic: those used to learn from the environment through tactile or haptic exploration

The semiotic function of gesture is to communicate meaningful information. The structure of a semiotic gesture is conventional and commonly results from shared cultural experience, thus their meanings are not required to be made explicit between performers. The good-bye gesture is an example. The ergotic function of gesture is associated with the notion of work. It corresponds to the capacity of humans to manipulate the real world, to create artifacts or to change object's position, orientation and shape according to our needs. The epistemic function of gesture allows humans to learn from the environment through tactile experience. By moving an hand over an object, one can figure out its structure, discover the material it is made of, as well as other properties. When interacting with robots semiotic gestures are the most relevant. Semiotic gestures can further be categorized according to their functionality. Table 2.1 illustrates a taxonomy of gesture categories which summarises all the following criteria.

A useful taxonomy of gestural types is one offered by [4]:

1. Iconic/ Manipulative gestures: convey information about the size, shape or orientation of an object
2. Pantomimic gestures: attempt to mimic an object or action
3. Symbolic gestures: have a precise meaning known by a group or a culture
4. Deictic gestures: pointing to a specific location
5. Abstract: gesture mapping is arbitrary

Table 2.1: Taxonomy of gestures

Nature	Manipulative	Gesture directly manipulates the object
	Pantomimic	Gesture imitates a real action
	Symbolic	Gesture visually depicts a sign
	Pointing	Gesture points to a specific location
	Abstract	Gesture mapping is arbitrary
Form	Static	No motion is in gesture
	Dynamic	Motion occurs in gesture
Temporal	Continuous	Action is performed during gesture
	Discrete	Action is performed after completion of gesture

This classification is related to the nature of the gesture and it describes the mapping of the gesture to the entities or to the intended task and how they relate to each other. For example a Manipulative gesture is when there is a direct mapping between the performed gesture and its impact on the object/entity, such as moving the arm up and down to move a quadrotor in the same directions, since the impact of the movements id directly related to the changing of the robot location.

In addition, a gesture can be dynamic or static. If the gesture includes motion or contains some types of dynamism while performing the gesture, in any of the body parts involved in the gesture, the gesture would be dynamic. Instead, a static gesture, or a posture, doesn't include any kind of motion or change while being performed.

Finally, according to the temporal dimension a gesture can be discrete or continuous. A gesture is classified as discrete if the impact of the gesture occurs after the gesture is completed. On the contrary, it is continuous if the recognition of the gesture is done while the gesture is performed, and user can see the impact of the gesture simultaneously. In particular, discrete gesture is when a gesture is performed to enter in a specific state or to do a specific action. After the recognition step, the system will respond to the gesture. For example, moving an hand in a circular manner to hover a drone. Instead a continuous gesture is when a user moves his arm to the left and right to navigate a robot to the left and right. As he moves his arm, the robot navigates to the same direction in real time.

The main challenge in human-robot interaction is to find a natural and intuitive communication modality able to provide an easy and fast way to interact so that, ideally, no learning or training is required. According to this, the selec-

tion of the gesture will influence both the performance of the system in terms of recognition accuracy, processing time, computational complexity and its usability. One way to create the gesture vocabulary is to leverage the concept of mental models and apply it to user interfaces for the human-robot interaction [5].

In [5], for example, they consider a gesture set to be coherent if all of its gestures adhere to one and the same metaphor. This metaphor evokes a certain mental model that, in turn, defines a certain behavior or , in the considered example, certain gestures. Selecting input commands according to a single metaphor promises to promote intuitive interaction. A system is considered intuitive if the way it works corresponds to our expectations. Thus, it should be fast to learn and easy to use. Mental models that define our expectations are formed by previously acquired knowledge and experiences. Peshkova et al. consider a gesture set to be intuitive if a single hint is enough to define all gestures in the set. So they clustered supportive examples of these gesture set models to navigate a UAV into three categories: *imitative* (direct mapping of the movements of operator to the vehicle motion), *instrumented* (an operator controls a vehicle through an imaginary intermediate link, which can be an imaginary physical object, such as a joystick), and *intelligent* (an operator assumes that a vehicle can interpret high-level multimodal commands). In [6], authors rely on gestures resembling actions performed in a real-world context of driving a car to navigate a wheeled robot. According to the guidelines reported in [7] they make this choice because the shape and kinematic model of wheeled mobile robots remind those of a car providing an implicit knowledge about robots behavior; since the actions required for driving a car are universal previous experience is not required. Moreover the control interface of a car, i.e., the set of actions and commands required to drive it, is among the ones with highest usability. The same context-driven choice of the gesture set appears to be made in [8], but they did not motivate and explain in details how they choose them. In [9] the authors introduce a set of multimodal commands and communication primitives suitable for the accomplishment of cooperative search tasks. In the proposed framework both command-based and joystick-based interaction metaphors can be exploited and smoothly combined to affect the robots behavior.

In [10] the proposed gesture vocabulary is coherent with a clean-up task in an office. The user has to cooperate with a robot to complete the task. The proposed gestures directly refer to common actions in cleaning task.

The above methods adopt a top-down approach to select gesture. Alternatively, is to use a bottom-up approach. This means that gesture are chosen on the basis of user preferences, thus they are defined as user-centered approaches. In this way the user is not asked to adopt an interaction method that has been already implemented, but on the contrary, the user-defined gestures can be used to develop a system that is suitable for the user needs. A clear and exhaustive ex-

ample of this approach is reported in [11]. The aim of this study is to investigate user-defined gestural interactions to navigate a drone. For each system action, different candidate gesture are proposed. The user has to perform each candidate gesture and rate it in terms of easiness. The gesture set is then composed by the gestures which have reached a high agreement score (all participants chose the same gesture for the same action) and having the highest occurrence. In addition, they point out that, all of the ten navigational actions were performed with dynamic deictic gestures and generally came with a high agreement between the users and a higher rating on how easy it was to think about them. This suggests that navigating a drone using the body can be a suitable modality to interact with a drone. A related study is conducted in [12]. Their user-centric design strategy aims at understanding how users naturally interact with drones. Users were asked to perform any action and interpret the proposed task freely. After each task, the participant was prompted to recall and explain their actions using a post-task think-aloud technique. Participants also rated their interaction in terms of suitability and simplicity. The gestures are then selected according to the same agreement score mentioned before. Moreover they found out gesture commands encompass the majority of interactions (86%) with respect to vocal commands (38%). Many other works in HRI fields using gesture input didn't focus on gesture vocabulary, but they just considered those that, from their point of view, have an intuitive coupling between command and effect [13], are "popular" [14], are based on common sense [15], are common in daily life [16], [17]. Most of them, stated that the proposed gestures are also easy to perform and to remember, but only few of this study reports results on usability tests (in particular [6],[13], [18],[9],[19],[20],[21],[10]), so it is difficult to prove their validity.

In [15],[19],[22],[23], the gesture are extracted from surveys or proposed on the basis of preliminary tests, but the user can personalize them or add new gestures.

By actively involving users in the creation of gesture sets to discover gestures with high consensus will be easier [7]. Therefore a further step to address natural interaction is users' personalization. From this point of view, a system is user-dependent if the gestures and their recognition are based on each user and they are involved in the training phase, while is user-independent if the final user does not have to train the system before using it.

Most of the current systems make a tradeoff between usability (naturalness) and recognition accuracy. On the one hand, in order to maximize the accuracy of gesture recognition, it is convenient to design a vocabulary of gestures easy to model and to implement. On the other hand, it is essential to use simple gestures, which are easier to perform and remember by users, and also yield lower computational complexity and faster response time.

2.2 Gesture Recognition Systems

According to Gartner's prevision made in July 2016 (Figure), gesture-based interfaces are becoming increasingly popular and they are supposed to reach a mainstream adoption in five to ten years. (frase+figura meglio nell'intro?)

Several approaches for gesture recognition can already be found in literature. Such methods can be generally classified into two main classes:

- *vision-based approaches*, which rely on one or more cameras to detect and analyze body movement from the video sequences. As opposite to other methods, this one allow the recognition of gestures remotely: the user does not have to wear any device. Thus is considered a non-intrusive method.

- *non-vision-based approaches*, which utilize devices that are physically in contact with the user. They are also known as contact-based methods [24].

Humans use their eyes to recognize gestures. For a robot a reasonable way to do the same is to use cameras to "see" the gesture. Vision-based approaches rely on image processing acquired with different systems, such as RGB cameras which provide two-dimensional visual informations or depth sensing input device which are able to capture gestures in 3D space. Many research activities in this field use Microsoft Kinect given the discriminative information provided by multimodal RGB-Depth data. Many recent contributions [18],[25],[15],[20],[10] consider Kinect-like sensors to extract gestures to control a robot, while in [26],[23],[27] the Kinect is used to evaluate the performances in terms of classification rate, accuracy and precision for the proposed algorithms. In detail, in [15] the Microsoft Kinect is used as gesture tracking device; recognized postures are then used to control a quadrotor. Users were asked for performing complete flight sessions (also called missions) from takeoff to landing. Experimental results using three different input devices show that the proposed Kinect solution reached a score of 100% completed missions while the solution using an iPhone interface reaches only 25% of mission completion. From these tests some useful information emerge for the usability of the system. Despite the fact that the use of Microsoft Kinect reaches the most accurate results, this comes at the expense of speed of completion of the mission. On the other hand, the use of the iPhone device was critical for those who have never used a multi-touch device [15]. Other vision-based systems use a single camera, that can be placed in a fixed point, or integrated with the robot [18],[27]. Other systems adopt a marker-based solution, like motion capture. Vision-based techniques have many drawbacks. They require controlled and uniform lighting conditions and proper camera angles. In addition, these methods suffer from high computation complexity [18], calibration difficulties, high energy requirement [18], a limited wearability and obtrusiveness.

Table 2.2: The main existing gesture recognition approaches pros and cons

Vision-based	Non-vision-based
dynamic light condition	infrastructure-less
camera angle	mobility
camera calibration	no light dependent
limited fov	low computation complexity
fixed distance from the camera	high precision
high computation complexity	wearability constraints
energy requirement	occlusion free
background noise	position independent
non-obtrusive	
no need to wear any device	

Other limitations can derive from total or partial occlusion [20]. Another issue can be the limited field of view and the fixed distance from the camera that has to be maintained in order to interact with the system [10]. Also the noise in image acquisition can derive from the background color. In [23],[28] a dedicated algorithm is used to perform background subtraction.

To overcome these limitations, physical interaction can be exploited to implement non-vision-based techniques. One of the most popular methods consists of recognizing gestures utilizing sensor gloves. In this regard, the authors of [29] exploits gloves to directly measure arm joint angles and hand movements using an integrated IMU sensor. Although they provide high precision in information acquisition, the user has to wear gloves limiting his freedom of motion. Another way to acquire gestures is to use touch screen technologies. In [15], gestures are acquired through a smartphone interface. Also this interface requires both hands and it can be not practical in many scenarios. Table 2.2 reports advantages and disadvantages of vision-based and non-vision-based techniques discussed above.

Relative novel solutions rely on data from accelerometers and gyroscopes acquired by wearable devices which are gaining in popularity. Pros and cons are summed up in Table 2.3. Wearable sensors are an interesting alternative to vision, since their effectiveness is related only to a minor extent to environmental conditions, and there are no constraints on the mutual spatial relationships between the robot and the person herself. Furthermore, the low requirements in terms of power consumption allow for a prolonged use of the device [6]. Previous works exploring this alternative to acquire inertial data typically use devices like MYO smartband [9], [21], smartwatch [6],[13],[14],[30],[31],[32],[17],[33], MEMs

Table 2.3: Wearable devices pros and cons

Smartwatch	Sensorized gloves
infrastructure-less	high precision
high mobility	no hands-free
low cost	high cost
haptic feedback	calibration
hands-free interaction	
non intrusive	
limited computational capabilities	

[34], smartphone [19],[35], smart belt [33] or other joystick-like device such as the well-known Wiimote [22],[36],[8],[37],[38] and so on. Only a few of them use these devices to control a robot, both flying vehicles [13],[9],[38] and ground vehicles [6],[19],[8],[33]. These devices have the advantage of being low cost, low power, compact and not obtrusive. In particular, a new generation of smartwatches is leading the way in wearable computing interfaces. These sensors can give relatively accurate motion and orientation information of users' hands at a high frame rate and can thus be used for gesture input. While data from camera-based sensors are prone to occlusion, and its quality of accuracy is highly dependent on the position of the user relative to the sensor as said before, IMU sensors are occlusion-free and position-independent. In addition, inertial data can be used with less complex processing compared with data from camera-based sensors.The disadvantage of IMU sensors is that they cannot capture hand shape information [23].

On these premises, it makes sense to develop an architecture for the gesture-based control of a robot, which relies on the inertial information provided by a sensor embedded in a commercial smartwatch. In [6], for example, a gesture-based architecture is proposed to control a wheeled robot, while in [13] to control a UAV. Both rely only on data acquired through a smartwatch and reach good results in terms of accuracy and system usability tested by making real trials.

2.3 Gesture Recognition Methods

The problem of gesture recognition can be generally divided in two sub-problems: (1) the gesture representation problem, i.e., how to create a model for each gesture and (2) the decision problem, i.e., how to classify a gesture. Independently of the adopted device and the gesture representation, several methods for inference can

be applied to gesture recognition.

2.3.1 Sensor and Feature Extraction

Gesture recognition from inertial data is generally preceded by a feature extraction step. The feature extraction stage is concerned with the detection of features used for the estimation of parameters of the chosen gestural model. A fully comprehensive overview on feature extraction process is reported in [39]. This process can be divided into three sub-steps: feature computation, feature selection and feature extraction. Signal characteristics such as time-domain and frequency-domain features are widely used for feature calculation. *Time-domain features* include mean, median, variance, skewness, kurtosis, while *frequency-domain features* include peak frequency, peak power, spectral power on different frequency bands and spectral entropy [39]. Time-domain features are widely used in the field of gesture recognition. Other time-domain features such as the Correlation Coefficient are also used in [13],[40]. One of the most important frequency-domain features used for human activity recognition is the Discrete Fourier Transformation (DFT). This feature has been used in [41] to recognize gesture's input for smartphones and smartwatches. The peak and valley frequency has been used in several studies related to gesture recognition, such as in [19]. Gesture energy , i.e the intensity of each gesture in the movement process, is another used feature [42]. The feature selection process is defined as a process of searching a subset of appropriate features from the original set. Feature selection is an important step in the use of machine-learning algorithms as it reduces computation time and complexity, while improving the overall classification rate. Liu et al. [43] categorize the feature selection process in a three-dimensional framework: a data mining task, an evaluation criterion, and a search strategy. According to [44] the feature selection process is generally categorized into three categories:

- Filters: it consists in computing the correlation between data and discard those which have a low correlation
- Wrappers: build a model and evaluate it by adding (forward) or removing (backward) features step-by-step. The search space of different subsets of features needs to be explored to determine the optimum combination of features.
- Embedded: the algorithm returns a sparse model and automatically chooses significant features.

The combination of original features is an alternative way of selecting a subset of relevant features. This technique consists in combining the original feature set

in order to define a new relevant features set. In other words, feature extraction is the transformation of high-dimensional data into a meaningful representation data of reduced dimensionality. The main advantage of feature extraction is that it facilitates classification and visualization of high-dimensional data. The most popular technique for feature extraction is Principal Component Analysis (PCA)[23], which is a linear technique that consists of transforming the original features into new mutually uncorrelated features. These new features are the so-called principal components. The main idea behind PCA is to remap the original features into a low dimensional space in which the principal components are arranged according to their variance. Another possible techniques is Random Projection (RP) proposed in [37] and used for dimensionality reduction.

2.3.2 Gesture Modeling

The selection of an appropriate gesture vocabulary is essential for achieving a further high classification rate. Many works just select randomly as template a sample for each performed gesture [22],[31],[8]. Others make some computations, like the average of N samples [9], [36], [17], or adopt a specific method like k-median clustering [32]. In [34] each gesture is encode in a sign sequence. Authors argue that the exact shape of the acceleration curves is not critical, but only the alternate sign changes of acceleration on x and z axes are required to uniquely differentiate any gestures. The work in [6],[45], propose a framework for the recognition of motion primitives, relying on Gaussian Mixture Modeling and Gaussian Mixture Regression for the creation of activity models. They adopt GMM and GMR to build 'expected curves' from examples of human motions in order to store the information in a compact way and, by creating models in the same space of the raw data, to make the comparison simpler and faster. It is worth to note that their system rely on two features only. A potentially better approach is adopting a RNN to create gesture model. A RNN is a special type of neural network that is able to handle both variable-length input and output. By training an RNN to predict the next output in a sequence, given all previous outputs, it can be used to model the joint probability distribution over sequences. RNNs consist of two parts: (1) a transition function that determines the evolution of the internal hidden state, and (2) a mapping from the state to the output. This expressive power and the ability to train via error backpropagation are key reasons why RNNs have gained popularity as generative models for richly-structured sequence data. RNNs model sequences by parameterizing a factorization of the joint sequence probability distribution as a product of conditional probabilities [46]. So far there has been few works that adopted a RNN to solve the problem of gesture recognition. Shin [47] proposed a model using wearable devices together with Long Short-Term Memory (LSTM) RNN. This algorithm applies the accel-

eration data directly to the RNN. The LSTM-based network was implemented using three linear units as input layer, one hidden layer with 128 neurons and the output layer with 8 softmax units corresponding to 8 target gesture movements. A 11.43% misclassification rate was reported for the LSTM-based model using acceleration data.

2.3.3 Gesture Classification

Gesture classification is the last but most important phase in gesture recognition process. Previous sections explained how a gesture is modeled, features are extracted from raw data, and the following sections discuss how extracted features are trained, classified and finally predicted. The features extracted/selected from the raw sensor data are used as inputs of the classification algorithms. In case of human activity recognition, the patterns of input data are associated with the gestures under consideration. In general, the classification task requires learning a decision rule or a function associating the inputs data to the classes. There are two main directions in machine learning techniques: supervised and unsupervised approaches [44]. Supervised learning approaches for classification such as Support Vector Machines (SVM), require entirely labeled activity data. The unsupervised learning approaches, such as those based on Gaussian Mixture Models (GMMs), Hidden Markov Models (HMMs) allow to infer automatically the labels from the data. Some of the most relevant classification techniques are discussed below.

- K-Nearest Neighbors (k-NN) is a supervised classification technique that can be seen as a direct classification method because it does not require a learning process. It just requires the storage of the whole data. To classify a new observation, the K-NN algorithm uses the principle of similarity between the training set and the new observation. The distance of the neighbors of an observation is calculated using a distance measurement called similarity function such as Euclidean distance. Moreover, one should note that when using the K-NN approach and a new sample is assigned to a class, the computation time increases as a function of the existing examples in the dataset [39]. In [9] the gesture classification can be directly obtained from an Euclidean distance: the accuracy achieved with this approach is 91,7%.

- Support Vector Machine (SVM) is a nonlinear supervised classifier. It takes a set of labeled feature vectors and creates a model that can be used to infer the class labels of unknown feature vectors. When building the model data might be mapped to a higher dimensional space using the kernel function to find a hyperplane that optimally separates the classes. In [14] is applied

SVM technique in combination with Global Alignment Kernel method to recognize smartwatch input gestures. The proposed approach provides some advantages in terms of computational complexity since its quadratic computational cost can be reduced, at the expenses of an approximate calculation, by tuning a single parameter. The recognition rate is 95,8%. In [48] the feature vectors prepared for the SVM sequentially contained Haar coefficients computed from the accelerometer signals x, y and z components. The confusion matrix shows that the average accuracy is 99.37%. In [35] a comparison between HMM classifier and SVM classifier is carried on. The SVM performs very well for small number of training samples, but it fails to improve as fast as the HMM. In case of eight training samples, the SVM algorithm results in 96% of accuracy on average.

- Random Forests (RF) consists of a combination of decision-trees. It improves the classification performance of a single-tree classifier by combining the bootstrap aggregating method and randomization in the selection of partitioning data nodes in the construction of decision tree. The assignment of a new observation vector to a class is based on a majority vote of the different decisions provided by each tree constituting the forest. However, RF needs huge amount of labeled data to achieve good performance. In [17], the authors propose a classification methodology to recognize, using acceleration data, different classes of motions, such as eating, drinking and other daily activities by comparing different machine learning techniques (Random Forests, Naive Bayes and MLP). The authors show that RF algorithm, for both personal and impersonal models, provides the highest average accuracy outperforming the other two methods.

- A Gaussian Mixture Model (GMM) is a probabilistic approach, generally used in an unsupervised classification. Unlike standard probabilistic models based on approximating data by a single Gaussian component density, GMM uses a weighted sum of finite Gaussian component densities. Using constructed features for human activity recognition, it is possible to learn separate GMMs for different gestures. Thus data classification can be performed, by selecting the GMM with the highest posterior probability. One of the drawbacks of this model is that in many cases the GMM does not guarantee the convergence to the global minimum. The GMM has been applied in several studies for human activity recognition, such as [6],[45]. In [6],[28],[49],[50],[45] GMM is exploited for the creation of activity models and provides an easy run-time recognition. In particular, in [49],[50] a modeling procedure based on Gaussian Mixture Modeling and Gaussian Mixture Regression has been chosen for three reasons: (i) it allows the cre-

ation of models of different resolution; (ii) the models can be projected in the space of run-time acceleration data, allowing for an easy run-time comparison; (iii) the models can be either person-specific or general-purpose, according to the chosen modeling dataset.

- A Markov chain represents a discrete time stochastic process covering a finite number of states where the current state depends on the previous one. In the case of gesture recognition, each gesture is represented by a state. A Markov chain is well adapted to model sequential data and is often used in a more general model that is the Hidden Markov Model (HMM). The HMM assumes that the observed sequence is governed by a hidden state sequence. Once the HMM is trained, the most likely sequence of activities can then be determined using the Viterbi algorithm [10]. As in the case of GMM, one of the drawbacks of HMMs is that in many cases this model does not guarantee the convergence to the global minimum. The HMM has been applied with good results in [20],[26],[23],[35]. In [20], Ad-hoc Hidden Markov Model are generated for each gesture exploiting a direct estimation of the parameters. Each model represents the best candidate prototype from the associated gesture training set. The generated models are then employed within a continuous recognition process that provides the probability of each gesture at each step.

- The DTW algorithm is a time-warping algorithm commonly used in real-time generated time series like motion and voice signal. Unlike Euclidean distance, this method tolerates offsets and time shifting during the comparison. It works as follows. First, for each class, one or more than one sequences are stored as template sequences. Then the test sequence is compared with the pre-stored template sequences, and the corresponding similarity score (or distance) for each template is computed. An important step in these comparisons is the alignment of the test sequence in time with each template sequence due to the variations in the sequence length. The time consumption of DTW is quite high, and it is one of the challenges to deal with. Since the DTW system has to maintain the template to match, as the amount of gestures to be recognized increases, the storage space required by the DTW algorithm will increase linearly. There are many attempts in improving DTW in different aspects [6], [18], [22],[36],[37], [31], [32], [8]. In [6],[50] is proposed a novel hybrid procedure based on Dynamic Time Warping and Mahalanobis distance which provides a method to compare 4-dimensional acceleration signals. Furthermore, the two techniques can be used independently (e.g., Mahalanobis distance alone) when real-time requirements must be met. In particular, in [6] the proposed method

reaches an accuracy of 93%. In [18] DTW approach, based on gesture specific features computed from depth maps, is adopted. In order to assure the fulfillment of the real time constraint, the DTW is executed in a multi-threaded way in which the different gestures are spread between different threads that run the gesture recognition method simultaneously, stopping in case one of the thread finds a gesture in the input sequence. The classification rate accuracy is reported for each gesture. In average the system reaches an accuracy of 70%. In [22], present uWave, an efficient recognition algorithm using DTW for such interaction using a single three-axis accelerometer. Unlike statistical methods, uWave requires a single training sample for each gesture pattern and allows users to employ personalized gestures. Authors introduce template adaptation in order to improve user-dependent recognition rate. uWave keeps two templates generated in two different days for each vocabulary gesture. As the user inputs more gesture samples, uWave updates the templates based on how old the current templates are and how well they match with new inputs. It shows that uWave achieves 98.6% accuracy in the user-dependent case. An improvement of the previous method is reported in [36],[37]. Both works employ DTW combined with compressive sensing to deal with the sparse nature of gesture in a user-independent context. The overall accuracy varies in a range between 96 % to 99 %. In [32] authors observed that different gestures have different quaternions, which is clearly visible in varying patterns of different dimensions of quaternion and these can be used for differentiating them. They use a hierarchical classifier. At first level classical DTW is applied on accelerometer and gyroscope signals. At second level,quaternions-based dynamic time warping (QDTW) is applied on quaternion time series to define a mapping between quaternion values of test shot and the quaternion values of existing templates set. This technique provides an efficient means for characterizing different arm/hand movements and gestures. The achieved accuracy is 90%.

- For classification step recent works tries to exploit neural networks capabilities. Different types of neural networks are employed. A short overview is presented in table 2.4. In [51] a Wiimote is used to capture accelerometer data. After a filtering and a normalization processing, k-means is performed to cluster trajectories of each gesture. Then a ANN is used to classify them. The reached accuracy is 92,00%. In [52] a continuous gesture recognition method is proposed. This approach is based on the idea of creating specialized signal predictors for each gesture class. These signal predictors forecast future acceleration values from the current ones. The errors between the measured acceleration of a given gesture and the predictors are used for

classification. These predictors are implemented using Continuous Time Recurrent Neural Networks (CTRNN). On the one hand, this kind of network exhibits rich dynamical behavior that is useful in gesture recognition and on the other hand, they have a relatively low computational cost that is an interesting feature for real time systems. In [53] a FNN is developed. One of the big advantages of FNN is that all learned knowledge is encoded in the weights, this makes the prediction very fast. A FNN consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one.The input gestures are classified directly by the feedforward neural network classifier in the case of basic gestures. Nevertheless, the input complex gestures go through an additional similarity matching. In particular, each basic gesture segmented from a complex gesture is encoded with a 4-bit Johnson code so that it is possible to calculate the similarity between gestures. The proposed recognition algorithm achieves an accuracy of 98,88%. In [54] authors present Binarized-BLSTM-RNN, a novel variation of the LSTM recurrent neural network with drastically reduced model sizes and memory demands. Thus they train a BLSTM-RNN with binary weights. This leads to a recognition accuracy of 81,00% which is a bit lower with respect to a standard LSTM-RNN, but it is faster in recognition. This method has also been compared with standard approaches, such as DTW and HMM, and it outperforms them in terms of classification rate and also in time response.

An overview of gesture recognition methods in terms of accuracy is shown in Table 2.5. As stated in [39] it is clear that comparing algorithm performance across different studies is a difficult task for many reasons. This difficulty is mainly related to: (i) the variability in the experimental protocols (the number of recruited subjects, the nature and the number of the recognized activities, the duration and the order of different activities, etc.); (ii) the type of sensors used (accelerometers, gyroscopes, etc) and their attachment to the body (wrist, chest, arm); (iii) the modeling and classification method adopted; (iv) the performance evaluation criteria (accuracy, F-measure, recall, precision, specificity, etc.).

Table 2.6 tries to highlight some relevant differences between the main recognition algorithm used for classification mentioned above.

2.4 Proposed Method and Contribution

Thus the main problems in gesture recognition emerging from the previous sections are to find a method which from the point of view of the user presents a fast response without sacrificing accuracy and precision, while on the system side

Table 2.4: Overview on methods using Neural Networks

Work	Devices	Input Sensor	Gesture Type	# Gesture	Features	Modeling	Classification	Accuracy	Year
[55]	Smartphone	Acc , gyr	Human activities	6	30, min tAcc/tGyr, max tAcc/tGyr, mean tAcc/tGyr, std tAcc/tGyr, mad tAcc/tGyr	-	NB K-NN SVM Softmax MLP	78,00% 89,00% 93,00% 92,00% 92,00%	2015
[51]	Wiimote	3 axis acc	circle triangle square Z	4	min acc x,y,z max acc x,y,z med acc x,y,z	filtering FFT normalization k-means	ANN	92,00%	2012
[52]	Acceelrometer	3 axis acc	circle triangle square infinity	8	-	CTRNN signal predictor	CTRNN	94,00%	2007
[47]	Smartwatch	3 axis acc	left/right up/down v s	8	-	RNN	RNN	24,00% misrec. rate	2016
[56]	ADLXL362	3 axis acc	left/right up/down Forw/ Backw On	7 complete wrapper PCA Relief-F TD,FD	24-8	-	ANN ELM-1 ELM-100 SVM	da 93,00% a 99,00%	2015
[53]	Pen-type	3 axis acc	left/right up/down Diag L,C,.. complex	24	25	FNN	encoding 4-bit + Similarity matching	98,88%	2016
[57]	Smartwatch	3 axis acc	activities	18	43	-	MLP	64,60%	2016
[58]	Kinect	-	Schaeffer language	25	11 Pose-based Angle-based Distance-based	-	LSTM RNN GRU	93,13% 90,21% 91,07%	2017
[59]	Myo Armband	9 axis inertial	safe op.	17	FD wavelet coeff	- -	FNN	88,00%	2016
[54]	3 different datasets	9 axis inertial	har	18	raw data TD TD	- -	B-BLSTM-RNN LSTM-RNN DNN DTW HMM	81,00% 83,20% 78,80% 73,30% 76,80%	2016

Table 2.5: Overview of recognition methods accuracy

W	Features	Modeling	Classification	Accuracy	UI
[6]	TD	GMM, GMR	Mahalanobis distance, DTW	93,95%	Yes
[13]	TD	Cross-Correlation	Correlation Coefficient	86,00%	Yes
[14]	TD	Gesture vocabulary	fGAK + SVM	95,80%	Yes
[34]	TD,FD	Sign sequence encoding	Template Matching	95,60%	Yes
[9]	TD	Average of samples	Euclidean distance	91,70%	Yes
[19]	FD	Kinematics-based	Peaks and valley detection	94,14%	Yes
[22]	TD	Random sample	uWave	98,6%	No
[36]	TD	TC+DTW+AF	DTW+CS	95,00%	Yes
[32]	TD	K-median clustering	DTW + QDTW	95,00%	Yes
[17]	TD	Average Random samples	RF	93,30%	No

has a low computational complexity and needs not too much training. Moreover since the feature extraction phase is crucial for the classification accuracy we need a technique which optimizes the feature selection. There are many algorithms to find relevant feature, such as backpropagation. One of the advantage of neural networks is that in many cases no data pre-processing is needed and raw data can be used directly.

As explained in [60] one possible solution might be to adopt RNN that can be draw as an unfolded computational graph, in which each component is represented by many different variables, with one variable per time step, representing the state of the component at that point in time. The unfolding process introduces two major advantages [60]:

- 1. Regardless of the sequence length, the learned model always has the same input size, because it is specified in terms of transition from one state to another state, rather than specified in terms of a variable-length history of states.

- 2. It is possible to use the same transition function with the same parameters at every time step.

These two factors make it possible to learn a single model that operates on all time steps and all sequence lengths, rather than needing to learn a separate model for all possible time steps. Learning a single, shared model allows generalization to sequence lengths that did not appear in the training set, and allows the model to be estimated with far fewer training examples than would be required without

Table 2.6: Comparison of the main recognition algorithms: DTW, GMM, HMM, SVM, RNN

DTW	**GMM**	**HMM**	**SVM**	**RNN**
- Storage space increase linearly	+ ability to trade off accuracy of representation against data volume	real-time classification	+ real time classification	- require a large amount of data for training
+ temporal alignment between sequences of different lengths	+ probabilistic model	- computational complexity proportional to the number and the dimension of the feature vectors	+ suited both for user-dependence and user-independence	+ real time constraint
+ limited training dataset	+ support models of different resolution	+ fixed storage space	+ class-discriminative model	+ fixed memory space
- high time cost	+ run time classification	- extensive training data needed	non linear kernel compatibility	- heavy computation
- heavy computation	+ suited both for user-dependence and user-independence	- restrict variation of gesture vocabulary	- time consuming with a large amount of data	- large memory
-not scalable	- not guarantee the convergence to a global minimum	+ capability in modeling spatio-temporal variability		- vanishing gradient
- doesn't support mix between templates	- not guarantee the convergence to a global minimum	statistical method		- slow training phase
		- observation depends only on the current state		+ generalization

parameter sharing. RNN should also solve the real-time constraint problem, since a gesture can be recognized faster with respect to a method using a template matching approach, since there is no need to wait a time windows, to make the classification, but it can be done in advance. Trivially less time we wait to classify a gesture and worst the accuracy will be.

Chapter 3

Second chapter

Summary

Examples of commonly used commands.

3.1 Algorithm

Chapter 4

Conclusions

Write the conclusions here...

Appendix A

Extra

Write here...

References

[1] D. Archer, "Unspoken diversity: Cultural differences in gestures," *Qualitative Sociology*, vol. 20, pp. 79–105, Mar 1997. 3

[2] C. Cadoz, *Les ralits virtuelles. Dominos. Flammarion.* 1994. 3

[3] J. L. Crowley and J. Martin, "Visual processes for tracking and recognition of hand gestures," in *In Workshop on Perceptual User Interfaces (PUI97*, 1997. 3

[4] B. Rim and L. Schiaratura, "Gesture and speech in fundamentals of nonverbal behavior," 01 1991. 3

[5] E. Peshkova, M. Hitz, and B. Kaufmann, "Natural interaction techniques for an unmanned aerial vehicle system," *IEEE Pervasive Computing*, vol. 16, pp. 34–42, Jan 2017. 5

[6] E. Coronado, J. Villalobos, B. Bruno, and F. Mastrogiovanni, "Gesture-based Robot Control: Design Challenges and Evaluation with Humans," 05 2017. 5, 6, 8, 9, 11, 13, 14, 18

[7] A. Malizia and A. Bellucci, "The artificiality of natural user interfaces," *Commun. ACM*, vol. 55, pp. 36–38, Mar. 2012. 5, 6

[8] X.-H. Wu, M.-C. Su, and P.-C. Wang, "A hand-gesture-based control interface for a car-robot," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 4644–4648, Oct 2010. 5, 9, 11, 14

[9] J. Cacace, A. Finzi, and V. Lippiello, "Multimodal interaction with multiple co-located drones in search and rescue missions," *CoRR*, vol. abs/1605.07316, 2016. 5, 6, 8, 9, 11, 12, 18

[10] S. Waldherr, R. Romero, and S. Thrun, "A gesture based interface for human-robot interaction," *Auton. Robots*, vol. 9, pp. 151–173, Sept. 2000. 5, 6, 7, 8, 14

[11] M. Obaid, F. Kistler, G. Kasparavičiūtė, A. E. Yantaç, and M. Fjeld, "How would you gesture navigate a drone?: A user-centered approach to control a drone," in *Proceedings of the 20th International Academic Mindtrek Conference*, AcademicMindtrek '16, (New York, NY, USA), pp. 113–121, ACM, 2016. 6

[12] J. R. Cauchard, J. L. E, K. Y. Zhai, and J. A. Landay, "Drone & me: An exploration into natural human-drone interaction," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, (New York, NY, USA), pp. 361–365, ACM, 2015. 6

[13] V. Villani, L. Sabattini, G. Riggio, C. Secchi, M. Minelli, and C. Fantuzzi, "A natural infrastructure-less human robot interaction system," *IEEE Robotics and Automation Letters*, vol. 2, pp. 1640–1647, July 2017. 6, 8, 9, 10, 18

[14] L. Porzi, S. Messelodi, and C. Modena, "A smart watch-based gesture recognition system for assisting people with visual impairments.," 10 2013. 6, 8, 12, 18

[15] A. Sanna, F. Lamberti, G. Paravati, and F. Manuri, "A kinect-based natural interface for quadrotor control," *Entertainment Computing*, vol. 4, no. 3, pp. 179 – 186, 2013. 6, 7, 8

[16] Y. Wu, K. Chen, and C. Fu, "Natural gesture modeling and recognition approach based on joint movements and arm orientations," *IEEE Sensors Journal*, vol. 16, pp. 7753–7761, Nov 2016. 6

[17] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 426–429, Feb 2016. 6, 8, 11, 13, 18

[18] G. Canal, S. Escalera, and C. Angulo, "A real-time human-robot interaction system based on gestures for assistive scenarios," *Comput. Vis. Image Underst.*, vol. 149, pp. 65–77, Aug. 2016. 6, 7, 14, 15

[19] W. Xian, P. T. Alonso, A. M. B. Barbolla, E. M. Moreno, and J. R. C. Corredera, "User-independent accelerometer-based gesture recognition for mobile devices," *Advances in Distributed Computing and Artificial Intelligence Journal*, vol. 1, pp. 11–25, December 2012. 6, 9, 10, 18

[20] S. Iengo, S. Rossi, M. Staffa, and A. Finzi, "Continuous gesture recognition for flexible human-robot interaction," in *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 4863–4868, May 2014. 6, 7, 8, 14

[21] G. C. Luh, H. A. Lin, Y. H. Ma, and C. J. Yen, "Intuitive muscle-gesture based robot navigation control using wearable gesture armband," in *2015 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1, pp. 389–395, July 2015. 6, 8

[22] J. Liu, L. Zhong, J. Wickramasuriya, and V. Vasudevan, "uwave: Accelerometer-based personalized gesture recognition and its applications," *Pervasive and Mobile Computing*, vol. 5, no. 6, pp. 657 – 675, 2009. PerCom 2009. 6, 9, 11, 14, 15, 18

[23] Y. Yin and R. Davis, "Real-time continuous gesture recognition for natural human-computer interaction," in *2014 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pp. 113–120, July 2014. 6, 7, 8, 9, 11, 14

[24] S. S. Rautaray and A. Agrawal, "Vision based hand gesture recognition for human computer interaction: A survey," *Artif. Intell. Rev.*, vol. 43, pp. 1–54, Jan. 2015. 7

[25] J. Nagi, A. Giusti, G. A. Di Caro, and L. M. Gambardella, "Human control of uavs using face pose estimates and hand gestures," in *Proceedings of the 2014 ACM/IEEE International Conference on Human-robot Interaction*, HRI '14, (New York, NY, USA), pp. 252–253, ACM, 2014. 7

[26] H. Abdirizak Abdullahi and B. Shafriza Nisha, "Gesture-based remote-control system using coordinate features," 04 2016. 7, 14

[27] M. Monajjemi, S. Mohaimenianpour, and R. Vaughan, "Uav, come to me: End-to-end, multi-scale situated hri with an uninstrumented human and a distant uav," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 4410–4417, Oct 2016. 7

[28] H. D. Yang, A. Y. Park, and S. W. Lee, "Gesture spotting and recognition for human ndash;robot interaction," *IEEE Transactions on Robotics*, vol. 23, pp. 256–270, April 2007. 8, 13

[29] K. A. Bhaskaran, A. G. Nair, K. D. Ram, K. Ananthanarayanan, and H. R. N. Vardhan, "Smart gloves for hand gesture recognition: Sign language to speech conversion system," in *2016 International Conference on Robotics and Automation for Humanitarian Applications (RAHA)*, pp. 1–6, Dec 2016. 8

[30] V. Villani, L. Sabattini, N. Battilani, and C. Fantuzzi, "Smartwatch-enhanced interaction with an advanced troubleshooting system for industrial

machines," *IFAC-PapersOnLine*, vol. 49, no. 19, pp. 277 – 282, 2016. 13th IFAC Symposium on Analysis, Design, and Evaluation ofHuman-Machine Systems HMS 2016. 8

[31] D. Moazen, S. A. Sajjadi, and A. Nahapetian, "Airdraw: Leveraging smart watch motion sensors for mobile human computer interactions," in *2016 13th IEEE Annual Consumer Communications Networking Conference (CCNC)*, pp. 442–446, Jan 2016. 8, 11, 14

[32] R. Srivastava and P. Sinha, "Hand movements and gestures characterization using quaternion dynamic time warping technique," *IEEE Sensors Journal*, vol. 16, pp. 1333–1341, March 2016. 8, 11, 14, 15, 18

[33] J. G. Lee, M. S. Kim, T. M. Hwang, and S. J. Kang, "A mobile robot which can follow and lead human by detecting user location and behavior with wearable devices," in *2016 IEEE International Conference on Consumer Electronics (ICCE)*, pp. 209–210, Jan 2016. 8, 9

[34] R. Xu, S. Zhou, and W. J. Li, "Mems accelerometer based nonspecific-user hand gesture recognition," *IEEE Sensors Journal*, vol. 12, pp. 1166–1173, May 2012. 9, 11, 18

[35] Z. Prekopcsk, "Accelerometer based real-time gesture recognition," 2008. 9, 13, 14

[36] A. Akl and S. Valaee, "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, compressive sensing," in *2010 IEEE International Conference on Acoustics, Speech and Signal Processing*, pp. 2270–2273, March 2010. 9, 11, 14, 15, 18

[37] A. Akl, C. Feng, and S. Valaee, "A novel accelerometer-based gesture recognition system," *IEEE Transactions on Signal Processing*, vol. 59, pp. 6197–6205, Dec 2011. 9, 11, 14, 15

[38] A. Kanso, I. H. Elhajj, E. Shammas, and D. Asmar, "Enhanced teleoperation of uavs with haptic feedback," in *2015 IEEE International Conference on Advanced Intelligent Mechatronics (AIM)*, pp. 305–310, July 2015. 9

[39] F. Attal, S. Mohammed, M. Dedabrishvili, F. Chamroukhi, L. Oukhellou, and Y. Amirat, "Physical human activity recognition using wearable sensors," *Sensors*, vol. 15, no. 12, pp. 31314–31338, 2015. 10, 12, 16

[40] D. Mellinger, N. Michael, and V. Kumar, "Trajectory generation and control for precise aggressive maneuvers with quadrotors," *The International Journal of Robotics Research*, vol. 31, no. 5, pp. 664–674, 2012. 10

[41] M. Xie and D. Pan, "Accelerometer gesture recognition," 2014. 10

[42] J. Xu, Y. L. Dong, and Y. Tang, "Gesture recognition based on wearable sensing," in *2016 Chinese Control and Decision Conference (CCDC)*, pp. 2763–2768, May 2016. 10

[43] A. M. Khan, M. H. Siddiqi, and S.-W. Lee, "Exploratory data analysis of acceleration signals to select light-weight and accurate features for real-time activity recognition on smartphones," *Sensors*, vol. 13, no. 10, pp. 13099–13122, 2013. 10

[44] C. C. Aggarwal, *Data Mining: The Textbook*. Springer Publishing Company, Incorporated, 2015. 10, 12

[45] B. Bruno, F. Mastrogiovanni, and A. Sgorbissa, "Recognition of human activities through wearable accelerometers," in *[workshop] Robot and Human Interactive Communication, 2014 RO-MAN: The 23rd IEEE International Symposium on*, 2014. 11, 13

[46] J. Chung, K. Kastner, L. Dinh, K. Goel, A. Courville, and Y. Bengio, "A recurrent latent variable model for sequential data," in *Proceedings of the 28th International Conference on Neural Information Processing Systems*, NIPS'15, (Cambridge, MA, USA), pp. 2980–2988, MIT Press, 2015. 11

[47] S. Shin and W. Sung, "Dynamic hand gesture recognition for wearable devices with low complexity recurrent neural networks," *CoRR*, vol. abs/1608.04080, 2016. 11, 17

[48] M. Khan, S. I. Ahamed, M. Rahman, and J. J. Yang, "Gesthaar: An accelerometer-based gesture recognition method and its application in nui driven pervasive healthcare," in *2012 IEEE International Conference on Emerging Signal Processing Applications*, pp. 163–166, Jan 2012. 13

[49] B. Bruno, F. Mastrogiovanni, A. Saffiotti, and A. Sgorbissa, *Using Fuzzy Logic to Enhance Classification of Human Motion Primitives*, pp. 596–605. Cham: Springer International Publishing, 2014. 13

[50] B. Bruno, F. Mastrogiovanni, A. Sgorbissa, T. Vernazza, and R. Zaccaria, "Analysis of human behavior recognition algorithms based on acceleration data," 05 2013. 13, 14

[51] B. M. Lee-Cosio, C. Delgado-Mata, and J. Ibanez, "Ann for gesture recognition using accelerometer data," *Procedia Technology*, vol. 3, pp. 109 – 120, 2012. The 2012 Iberoamerican Conference on Electronics Engineering and Computer Science. 15, 17

[52] G. Bailador, D. Roggen, G. Tröster, and G. Triviño, "Real time gesture recognition using continuous time recurrent neural networks," in *Proceedings of the ICST 2Nd International Conference on Body Area Networks*, BodyNets '07, (ICST, Brussels, Belgium, Belgium), pp. 15:1–15:8, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007. 15, 17

[53] R. Xie and J. Cao, "Accelerometer-based hand gesture recognition by neural network and similarity matching," *IEEE Sensors Journal*, vol. 16, pp. 4537–4545, June 2016. 16, 17

[54] M. Edel and E. Kppe, "Binarized-blstm-rnn based human activity recognition," in *2016 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pp. 1–7, Oct 2016. 16, 17

[55] Z. Wu, S. Zhang, and C. Zhang, "Human activity recognition using wearable devices sensor data," 2015. 17

[56] G. Marqus and K. Basterretxea, "Efficient algorithms for accelerometer-based wearable hand gesture recognition systems," in *2015 IEEE 13th International Conference on Embedded and Ubiquitous Computing*, pp. 132–139, Oct 2015. 17

[57] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, pp. 426–429, 2016. 17

[58] S. Oprea, A. Garcia-Garcia, J. Garcia-Rodriguez, S. Orts-Escolano, and M. Cazorla, "A recurrent neural network based schaeffer gesture recognition system," in *2017 International Joint Conference on Neural Networks (IJCNN)*, pp. 425–431, May 2017. 17

[59] A. Srisuphab and P. Silapachote, "Artificial neural networks for gesture classification with inertial motion sensing armbands," in *2016 IEEE Region 10 Conference (TENCON)*, pp. 1–5, Nov 2016. 17

[60] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org. 18

[61] S. Fu, H. Saeidi, E. Sand, B. Sadrfaidpour, J. Rodriguez, Y. Wang, and J. Wagner, "A haptic interface with adjustable feedback for unmanned aerial vehicles (uavs) -model, control, and test," in *2016 American Control Conference (ACC)*, pp. 467–472, July 2016.