Characterizing human grasping movements by means of an acceleration sensor

FERNANDO DEL CAMPO
Universidad Diego Portales
School of Information Engineering
Ejército 441, Santiago
CHILE
fernando.delcampo@mail.udp.cl

MIGUEL CARRASCO
Universidad Diego Portales
School of Information Engineering
Ejército 441, Santiago
CHILE
miguel.carrasco@udp.cl

Abstract: A majority of systems that take advantage of human motion in order to recognize gestures are developed through temporal image processing algorithms. However, thanks to the increasing development of acceleration sensors in recent years, it has become possible to use actual arm movements as an acquisition system. This feature could be used in more intuitive systems to communicate reach-to-grasp movements. This research proposes placing an accelerometer on a user's arm to recognize grasping movements in an unique way. The most complex part of this problem revolves around the fact that an accelerometer is unable to evaluate whether a user is performing an reach-to-grasp movement. Given that the movement involves a temporary action, it is possible to use a hidden Markov system to dynamically predict user grasping movements. The results indicate that the model can correctly predict all movements with an F-score = 99% on average.

Key–Words: Hidden Markov Model, Human Computer Interaction, Gesture recognition, Accelerometers, Grasping movements

1 Introduction

Human movement analysis is an area of study that has been quickly expanding over the past few years. Progress in analyzing image sequences, the evolution of computer systems and the miniaturization of technology used to capture movement have made motion analysis applications possible in areas such as athletics, psychology, therapies, military and security systems, prevention in high risk dangerous areas, customer movement statistics in retail outlets (e.g. [1, 2, 3]) . On the human gesture level, the majority of research has been conducted around the analysis of gesture language, of which sign language is the most common. However, communicative gestures are only one small part of all the gestures made by humans.

Out of the thousands of human gestures that exist, this paper focuses on grasping movements. These gestures have not been studied extensively because movements that occur in natural human interaction with their surroundings are difficult to capture in a controlled environment. Therefore, within the system parameters it is necessary to include perception and knowledge of the environment. Before even considering body segmentation and motion recognition in image sequences, the system must solve many inherent difficulties. These challenges include signal variability over time: direction and dynamics of gestures vary with each execution even when performed by the same person; signal variability in space: similarly to varying during execution, movements also vary in each dimension of space; temporal or macro segmentation: it is necessary to segment each sequence temporally to effectively study each individual gesture. Every human gesture belongs within a particular context. Grasping movements are linked to an object's intrinsic nature (shape, size) as well as its extrinsic characteristics (position and orientation relative to the user) [4]. Therefore, accurately defining specific gestures and the decision process (using macro segmentation) then recognizing those movements constitutes determining an important focus of study.

To avoid the effects of image sequence analysis, this investigation aims to measure arm acceleration movements through devices incorporated directly on a users body, specifically a Wii (Wiimote) attached to the arm (see Fig.1). It is important to stress that the acceleration sensor cannot recognize grasping movements given that a mechanism to measure grasping motions does not yet exist. Therefore, the current goal is to characterize a few specific types of arm movements in order to recognize exactly when a person intends to grasp an object. Because the action involves a temporary movement, a hidden Markov

model (HMM) is used to determine which specific action is being performed. The greatest complication of this system is formulating features that the Markov system can assess. In order to accomplish this formulation, a set of extracted features is created from the acceleration sensor which makes it possible to predict which movements the grasping action involves with high probability.

2 Background

One relevant area of the human-computer interaction is the gesture recognition. Normally people use arm and hand gestures to do actions like grasping, playing, writing, painting, etc. Nowadays, gesture recognition is being used in many areas. Kim et al. [5] proposed gesture recognition utilizing four processes; hand detection and tracking, extracting a meaningful gesture from an image sequence, specific feature extraction and finally, gesture recognition.

Motion detection utilizing a camera is not a trivial task due to lighting changes and complex backgrounds in addition to tracking quick hand movements [6]. To solve the hand tracking problem, techniques such as the condensation algorithm [7] or Bayesian networks [8] have been used. However, the largest obstacle is the high computation time. Another current problem in the field is gesture segmentation, i.e., how to distinguish when one movement ends and another begins. The variety of patterns in a gesture, compared to the total number of possible gestures, creates a difficult feature extraction and gesture recognition process. Hidden Markov Model applications or neural networks are typically used to recognize movement patterns (e.g. [5]).

Hidden Markov Model: Currently there are a few different gesture recognition systems based on input devices with sensors such as Wii controls and HMM models as inference algorithms for gesture recognition [9]. Among the developed systems is the Schlomer, et al. [9] application which can be trained with user defined movements, meaning it is not limited by predetermined gestures. Likewise, Han et al. [10] has developed systems that are capable of detecting normal daily motions such as walking, climbing stairs, running and even falling among others. These systems show that the acceleration sensor has the potential to recognize and monitor daily activities automatically. One of the major difficulties facing the acceleration sensor is the positioning that is required when constructing system parameters. To deal with this effectively, Han et al. [10] developed a method that compensates for device rotation, independent of sensor position, through an HMM model. More relevantly, video games have increasingly used this technology, providing interactive systems which operate with acceleration sensors [11]. In addition, new mobile phones increasingly employ sensors to control games and applications. This is the case for the framework developed in [12] which uses a touch screen and accelerometer as the interface for video games as well as an a HMM system for gesture recognition. In this same category are applications that use built in accelerometers, such as in the iPhone, which have been used successfully in gesture recognition (e.g. [13]).

3 Proposed Method

This section describes implemented solutions and addresses (I) feature construction, (II) model construction, (III) training and simulation.

(I) Feature Construction. This phase aims at building the feature vector needed to construct the HMM model. The first step is data preprocessing. In order to do this, a temporal window is determined which allows the system to obtain the largest quantity of data whilst still preserving movement history. The results obtained with a Wii sensor indicate that this range corresponds to time window 0.4 (s). By utilizing this window size, all intervals have sufficient measurements to reduce noise that could be present and additionally, no window is left without data to process. This resolves the data preprocessing without requiring any additional algorithms (Fig. 2).

For each window of time ($\mathbf{T}_i$) a feature vector is constructed with the average mean values from each window. It was necessary to determine a new feature set to improve the HMM model based on the previous signals. These operations were the positive scalar sum of the three acceleration axes (1), angle calculation using Pythagorean theorem (2) and the differences in acceleration measurement and the angle between time $t(n)$ and $t(n+1)$.

$$Acc = \sqrt{{Acc_X}^2 + {Acc_Y}^2 + {Acc_Z}^2} \tag{1}$$

$$\alpha = \arctan\left(\frac{\sqrt{{Acc_Z}^2}}{\sqrt{{Acc_X}^2}}\right) \tag{2}$$

$$\beta = \arctan\left(\frac{\sqrt{{Acc_X}^2}}{\sqrt{{Acc_Y}^2}}\right) \tag{3}$$

$$\gamma = \arctan\left(\frac{\sqrt{{Acc_Y}^2}}{\sqrt{{Acc_Z}^2}}\right) \tag{4}$$

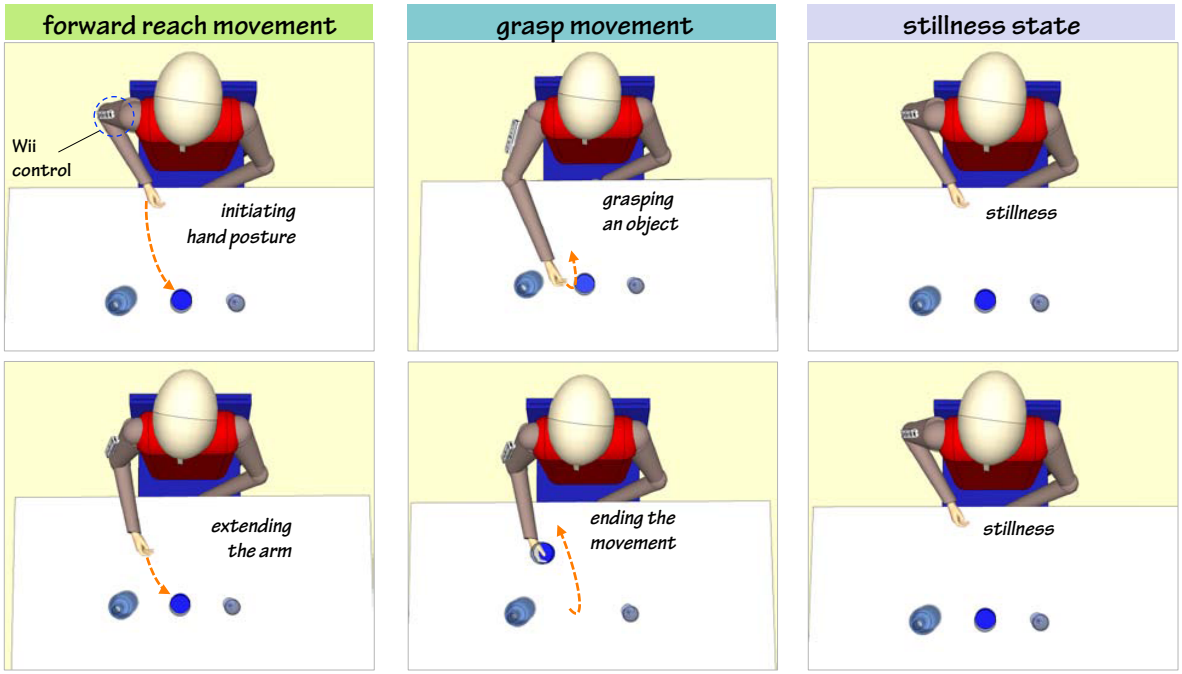

Figure 1: General Framework. Three movements states detected by our system.

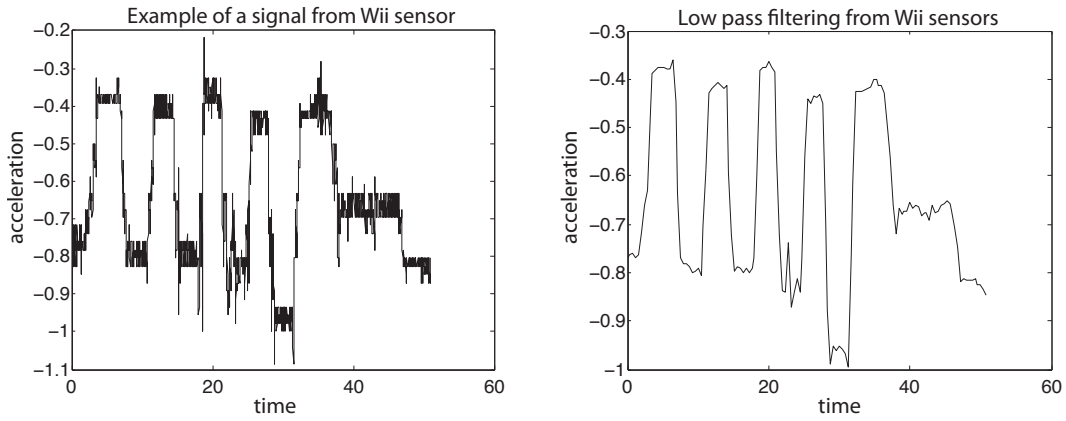

Figure 2: Data preprocessing.

$$\Delta F(t) = F(t) - F(t-1)$$
$$\text{for all } F = \{Acc_X, Acc_Y, Acc_Z, Acc, \alpha, \beta, \gamma\} \quad (5)$$

Each movement has an associated feature vector, the following four steps are performed in order to construct the vector: (1) Determine the start and end point of each movement according to class. (2) Assign each feature vector to a class. This is determined by measuring a movements start and end times, then recording and distinguishing what kind of model class the feature vector pertains to. (3) Move time measurements to the source to ensure that the feature vector is independent from the time measurement, but not the event. For this reason, we calculated the average mean of each signal.

$$\overline{\mathbf{x}} = \frac{1}{n}\sum_{i=0}^{n} \mathbf{T}_i$$

Using this procedure, each signal is calculated and displaced by the same amount towards the origin. (4) In order for feature models to consider the sequence of events, they are converted from a set of feature vectors belonging to the same class, to a feature vector that represents them all. Each feature vector represents a point on a line in time. Considering this, an nth grade polynomial is reconstructed so that it represents all points. Now time becomes part of the feature vector, but not as a field, it is internalized in the values

of each field. At the same time, it is just one vector that represents the action. To construct this equation, a 3rd degree polynomial interpolation is used taking advantage of the least squared method. This replaces all pertinent measurements within the same motion for a single feature. For each feature, four values are generated which represent the feature $\{\mathbf{A}_3, \mathbf{A}_2, \mathbf{A}_1, \mathbf{A}_0\}$.

$$\mathbf{P}\big(x\big) = A_3x^3 + A_2x^2 + A_1x + A_0 \qquad (6)$$

The design of these features provides more information than the signal in itself, this means that the polynomial parameters contain all points along the motion (Fig.3).

(II) Construction of the HMM model: The HMM model was designed with three states. These states are reach movement, retreat movement and stillness state. The observations that were defined above are the four components of polynomial functions of acceleration and its possible combinations of axes, angles as well as differences in acceleration and angle. This is defined to identify the best combinations to re-learn a movement. The probabilities of transition between states and the probabilities of an observation sequence were obtained through the Baum-Welch algorithm [14] which estimates the transition probabilities, in order to find emission probabilities, given a sequence of observations using the algorithm Forward-Backward [14]. The proposed feature allows us to analyze the way in which each feature is relevant to classify a particular movement.

$$\begin{aligned}
h_1 &= \mathbf{P}(Acc_X) = [A_3A_2A_1A_0]_{Acc_X} \\
h_2 &= \mathbf{P}(Acc_Y) = [A_3A_2A_1A_0]_{Acc_Y} \\
h_3 &= \mathbf{P}(Acc_Z) = [A_3A_2A_1A_0]_{Acc_Z} \\
h_4 &= \mathbf{P}(\alpha) = [A_3A_2A_1A_0]_{\alpha} \\
h_5 &= \mathbf{P}(\beta) = [A_3A_2A_1A_0]_{\beta} \\
h_6 &= \mathbf{P}(\gamma) = [A_3A_2A_1A_0]_{\gamma} \\
h_7 &= \mathbf{P}(\Delta Acc_X) = [A_3A_2A_1A_0]_{\Delta Acc_X} \\
h_8 &= \mathbf{P}(\Delta Acc_Y) = [A_3A_2A_1A_0]_{\Delta Acc_Y} \\
h_9 &= \mathbf{P}(\Delta Acc_Z) = [A_3A_2A_1A_0]_{\Delta Acc_Z} \\
h_{10} &= \mathbf{P}(\Delta\alpha) = [A_3A_2A_1A_0]_{\Delta\alpha} \\
h_{11} &= \mathbf{P}(\Delta\beta) = [A_3A_2A_1A_0]_{\Delta\beta} \\
h_{12} &= \mathbf{P}(\Delta\gamma) = [A_3A_2A_1A_0]_{\Delta\gamma} \\
h_{13} &= \mathbf{P}(Acc) = [A_3A_2A_1A_0]_{\Delta Acc}
\end{aligned} \qquad (7)$$

Each feature signal is then transformed by an array of four feature polynomial factor by using the method describe in the previous step. In order to increase the number of features proposed, we formulate four new features composed by a lineal combination of the some previously proposed features.

$$\begin{aligned}
h_{14} &= h_1 \cup h_2 \cup h_3 \\
h_{15} &= h_4 \cup h_5 \cup h_6 \\
h_{16} &= h_7 \cup h_8 \cup h_9 \\
h_{17} &= h_{10} \cup h_{11} \cup h_{12}
\end{aligned} \qquad (8)$$

(III) HMM model training and simulation: In the training phase measurements of a set of previously classified data were randomly selected. These sets are exclusive, 70% correspond to training and the remaining 30% are used for testing. The training algorithm receives as input parameters all training observations, the total amount of those observations, the number of model states, the number of iterations of the Baum-Welch algorithm and finally tolerance, which is the finishing criterion estimated by the change in the Likelihood registry. The tolerance used for construction was 10^{-3} and only 10% of the measurements did not reach convergence within 20 iterations. Most of the data did so in less than 10 iterations. Once the model is trained, results are obtained as the transition values and emission. In the simulation phase, given a sequence of movements, the Forward-Backward algorithm is used followed by obtaining the likelihood of each class. After, which class corresponds to the movement through the maximization of Likelihood is determined.

$$\begin{aligned}
lik(h|\Theta) = \sum_{i=1}^{N} log \sum_{k=1}^{K} \pi_k p(h_i|\Theta_k) \\
\text{for all } h_i \text{ with } i = 1 \ldots N
\end{aligned} \qquad (9)$$

$$\Theta^*_{ML} = arg\ max_{\Theta} lik(\Theta) \qquad (10)$$

4 Experiments

In the experimentation phase multiple arm movements going towards an object were captured. The accelerations that were captured were then used in the evaluation of multiple HMM models. The gestures were captured by placing different objects on a table at a distance of 30-40 (cm) and having a user repeat grasping motions towards the objects. The test consists of approximately 50 movements which belong to each class. To determine strength of the constructed model,

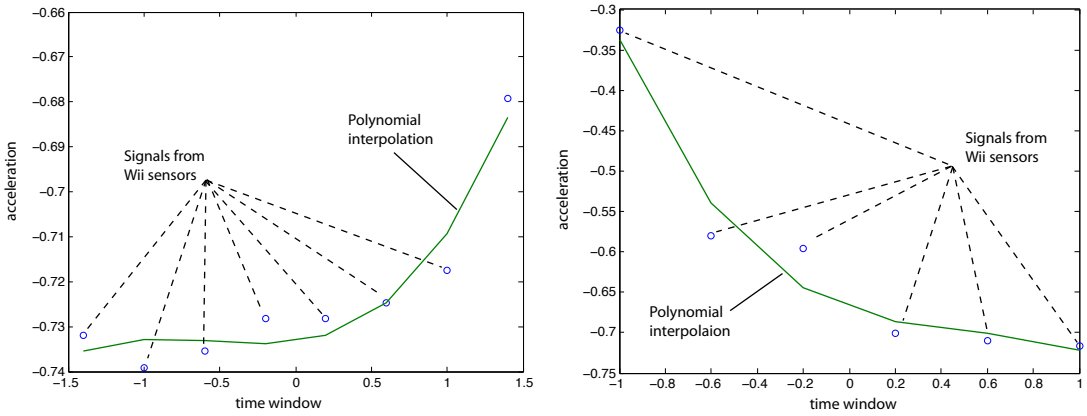

Figure 3: Polynomial reconstruction using 6 and 8 points of Wii accelerations data.

the precision estimates ($P = \frac{TP}{TP+FP}$) and sensitivity ($R = \frac{TP}{TP+FN}$) were evaluated. The variables of these equations correspond to values $\{TP, FP, FN\}$, which are equivalent to True Positive, False Positive and False Negative. In particular, precision (P) and sensitivity (R) should tend to 100% if classification were perfect. With these results, the performance was measured using the F-score statistics equation [15]. This is interpreted as the harmonic mean between precision and sensitivity for $\mathbf{Fscore} = \frac{2 \times P \times R}{P+R}$.

Using the above metric, a classification table was constructed to evaluate the performance for each simulation. In the best cases, the table has only numbers in diagonal. After constructing the tables, precision as well as sensitivity and finally, the Fscore, are calculated. Because these results depend heavily on the selection of training and simulation data, the Fscore is calculated using an average of 1000 iterations for each set of features. The results of each feature evaluation according to the type of movement are shown in Fig.4

The characteristics which obtained the best results correspond to h11 and h14. Starting with these results, the feature combinations which resulted in the model with better average yield are trained, shown in Fig.5.

5 Conclusions

Through the experiments explained above, this investigation has shown that it is possible to detect human gestures in reach-to-grasp tasks using an acceleration sensor, obtaining an F-score average of 99%. The main contribution of this work lies in the design of a features set that reflects human movement in reach-to-grasp tasks through a Markovian system. It is important to note that the acceleration sensor measurements provided by the Wii control cannot build an intent detection model without a prior transformation. This paper proposes to transform the temporal acceleration of polynomial feature sets.

Regarding the position of the Wii on the body, it was determined that the best configuration is one that moves in a normal direction towards the ground (*y-axis*), specifically held on the outside of the humerus bone. Thus, the moment the grasping movement is being acted out, the *y-axis* perceives more differences with respect to the horizon. This configuration was selected because it retains more information of the arm movement when the user is performing a grasping movement. A future work remains such as incorporating more types of movements into the model and increasing the number of features extracted by adding more acceleration sensors to the body arm, and incorporating image video sequences.

Acknowledgements: This work was supported by the National Commission of Science and Technology (CONICYT, Chile). Fondecyt grant no. 11100098 and from the School of Information and Telecommunication Engineering at Universidad Diego Portales.

References:

[1] R. Poppe, "Vision-based human motion analyis; an overview," *Computer Vision and Image Understanding*, no. 108, pp. 4–18, 2007.

[2] R. Poppe, "A survey on vision-based human action recognition," *Image and Vision Computing*, 2010.

[3] L. Sigal and M.-J. Black, "State of art in image- and video-based human pose and motion estima-

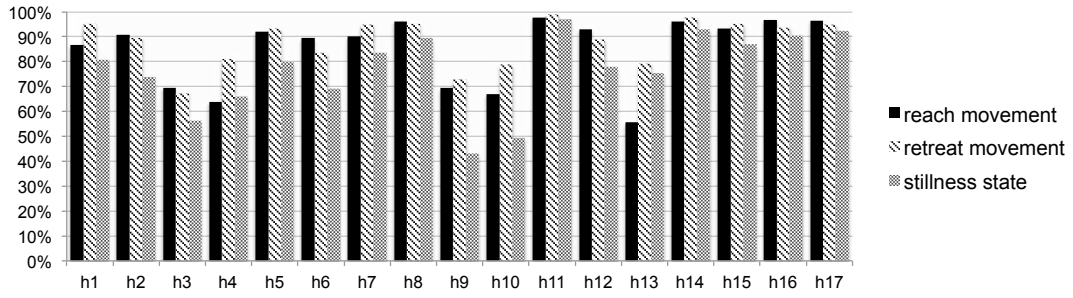

Figure 4: Average F-score of 17 features.

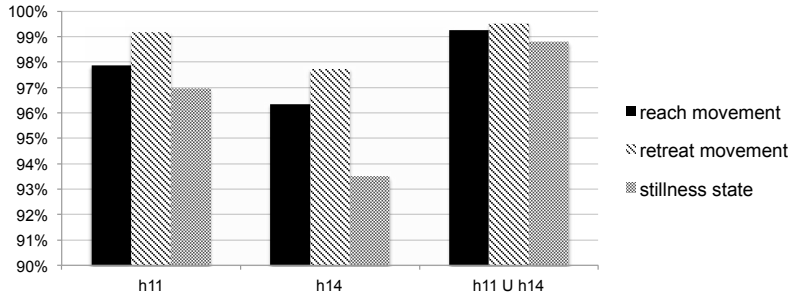

Figure 5: Best F-score averages of combined features.

tion," *International Journal on Computer Vision*, vol. 1, no. 87, pp. 1–3, 2010.

[4] S. El-Khoury, *Approche mixte, analytique et par apprentissage, pour la synthèse d'une prise naturelle*. PhD thesis, Université Pierre et Marie Curie, France, December 2008.

[5] K. K. Kim, K. C. Kwak, and S. Y. Chi, "Gesture analysis for human-robot interaction," in *The 8th International Conference on Advanced Communication Technology*, pp. 1824–1827, 2006.

[6] A. Shamaie and A. Sutherland, "A dynamic model for real-time tracking of hands in bimanual movements," in *LNCS*, vol. 2915, pp. 172–179, 2004.

[7] M. J. Black and A. D. Jepson, "A probabilistic framework for matching temporal trajectories: Condensation-based recognition of gestures and expressions," in *Computer Vision — ECCV'98*, vol. 1406 of *LNCS*, pp. 909–, 1998.

[8] J. Zhang, K. Yue, and W. Liu, "Learning and inferences of the bayesian network with maximum likelihood parameters," in *Advanced Data Mining and Applications*, vol. 5139 of *LNCS*, pp. 391–399, 2008.

[9] T. Schlomer, B. Poppinga, N. Henze, and S. Boll, "Gesture recognition with a wii controller," in *Proceedings of the Second International Conference on Tangible and Embedded Interaction (TEI'08)*, pp. 11–14, ACM, 2008.

[10] C. W. Han, S. J. Kang, and N. S. Kim, "Implementation of hmm-based human activity recognition using single triaxial accelerometer," *IEICE TRANS. FUNDAMENTALS*, vol. E93-A, pp. 1379–1383, July 2010.

[11] J. Liu, Z. Pan, and X. Li, "An accelerometer-based gesture recognition algorithm and its application for 3d interaction," in *ComSIS, Special Issue*, vol. 7, pp. 177–188, 2010.

[12] M. Joselli and E. Clua, "grmobile: A framework for touch and accelerometer gesture recognition for mobile games," *VIII Brazilian Symposium on Games and Digital Entertainment*, pp. 141–150, 2009.

[13] M. Klingmann, "Accelerometer-based gesture recognition with the iphone," Master's thesis, Goldsmiths University of London, September 2009.

[14] B. H. Juang and L. R. Rabiner, "Hidden markov model for speech recognition," *Technometrics*, vol. 33, no. 3, pp. 251–272, 1991.

[15] D. L. Olson and D. Delen, *Advanced Data Mining Techniques*. Springer, 2008.