Entertainment Computing 4 (2013) 11–24

Contents lists available at SciVerse ScienceDirect

Entertainment Computing

journal homepage: ees.elsevier.com/entcom

3D Gesture classification with linear acceleration and angular velocity sensing devices for video games

Salman Cheema *, Michael Hoffman, Joseph J. LaViola Jr.

University of Central Florida, Department of EECS, Orlando, FL 32816, USA

A R T I C L E I N F O

Article history:
Received 21 May 2012
Revised 3 July 2012
Accepted 23 September 2012
Available online 10 October 2012

Keywords:
Input devices
Interaction techniques
3D Gesture recognition
Machine learning
Video games
Linear Classifier
Support Vector Machine
AdaBoost
Decision Tree
Bayes Classifier

A B S T R A C T

We present the results of two experiments that explore various aspects of 3D gesture recognition using linear acceleration and angular velocity data. We examine relationships between variables affecting recognition accuracy, including size of gesture set, amount of training data, choice of classifier, and training configuration (user dependent/independent). Using a set of 25 gestures, we first compare the performance of four machine learning algorithms (AdaBoost, SVM, Bayes and Decision Trees) with existing results (Linear Classifier). Next, we investigate how results in existing literature apply to an application-oriented setting. We created a new 3D gesture database comprising 17,890 samples, containing examples of gestures performed in two different settings (a simple data collection setting vs a video game). We then compared the performance of all five classifiers on this new 3D gesture database. Our results indicate that the Linear Classifier can recognize up to 25 gestures at over 99% accuracy when trained in a user dependent configuration. However, in the video game setting, factors such as in-game stress and the ability to recall gestures cause a drop in recognition accuracy to 79%. We present a discussion of possible strategies to improve recognition accuracy in realistic settings by using a combination of recognition algorithms.

1. Introduction

A three-dimensional (3D) gesture can be defined as a motion traced in the air using an appropriate input device. Traditionally, 3D gestures have been used for interaction in virtual and augmented environments, specifically for tasks such as navigation, selection, manipulation, and system control [1]. More recently, advances in input technology have resulted in the widespread availability of commercial devices that incorporate 3D motion sensors such as accelerometers and/or angular rate gyroscopes. Examples of such devices include smartphones and video game controllers such as the Nintendo Wii Remote and the PlayStation Move. These sensors usually provide 3D input data in the form of linear acceleration and angular velocity. Using this data for gesture recognition can be a challenging task. There is no specific frame of reference, and the data can suffer from compounding errors due to drift. Different movement rates from different users can produce different acceleration and/or velocity profiles for a given gesture.

Additionally, software developers need to make informed decisions about the use of 3D gestures in an application during the design stage. The first decision is to select a set of gestures and to map them to specific tasks. The next step is to have a recognition system in place that can provide high recognition accuracy for the chosen gesture set. Depending on the nature of the application, recognition may be performed online or offline. The recognition system must be able to deal with tiny variations in the properties of individual gestures as users will not make precisely the same motion each time they perform a gesture. Examples of 3D gestures are therefore needed to train the recognition system. The amount of training data needed is often dependent on how many gestures there are in the gesture set, as well as whether user dependent or user independent gesture recognition is required. Application requirements may also place constraints on the amount of training data that can be collected.

In this work, we attempt to better understand the nature of 3D gesture recognition when using linear acceleration and angular velocity data. Specifically, we examine the following questions:

Q1. What is a good recognition algorithm to use? We compare accuracy across five different machine learning algorithms in this work. The algorithms chosen are a Linear Classifier based on Rubine's Algorithm [2], Decision Trees, Bayes Networks, Support Vector Machines and an AdaBoost Classifier with Decision Trees as the weak learner.

* Corresponding author. Tel.: +1 4072570444.
E-mail addresses: salmanc@cs.ucf.edu (S. Cheema), mhoffman.ucf@gmail.com (M. Hoffman), jjl@eecs.ucf.edu (J.J. LaViola Jr.).

1875-9521/$ - see front matter © 2012 International Federation for Information Processing Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.entcom.2012.09.002

Q2. How does varying the amount of training data affect recognition accuracy in user dependent and user independent configurations?
Q3. What is the largest subset of the 3D gestures in our experiments that can be recognized with high accuracy while using minimum amount of training data?
Q4. Does the setting of the application have an impact on the recognition accuracy? We collected gesture samples from participants in two different settings (a simple data collection setting vs a video game setting) and compared recognition accuracy between the two settings. In the data collection setting, users performed each gesture in a repetitive manner, before moving on to the next gesture, while in the video game, gestures were used to perform tasks in the game world.

To investigate the questions postulated above, we first compared the performance of our chosen machine learning algorithms on the 3D gesture dataset constructed by Hoffman et al. [3]. Secondly, we constructed a new database comprising 17,890 gesture samples performed with the Nintendo Wiimote. Our new database contains gesture samples acquired in both a simple data collection setting and in a video game setting. Although our findings are based on data collected with the Nintendo Wiimote, it should be noted that the features of data used in our analysis do not depend on the device. The features used in our investigation are derived from the nature of the 3D input data, i.e., linear acceleration and angular velocity. Therefore, we believe our findings are useful for constructing applications on any device or platform equipped with accelerometers and gyroscopes.

2. Related work

3D gesture recognition in virtual and augmented environments has been an important research topic over the years. Although there has been a significant amount of work on recognizing 3D gestures using traditional position and orientation tracking devices [4–7], the use of accelerometer and gyroscope-based devices for 3D gesture recognition has been sparse. Beedkar and Shah experimented with classifying 4 gestures using a Hidden Markov Model (HMM). The accelerometer data was gathered using numerous TAIYO SPP Bluetooth Accelerometer devices attached to the hands and feet of the participants. They concluded that 25 training samples per gesture were needed in order to achieve 96% accuracy [8]. Similarly, Kratz et al. [9] attempted to use HMMs to recognize 5 gestures in a video game setting. The experiments used 1 to 40 training samples comprising accelerometer data from a Wiimote as input for the classifier. The results showed a significant overall increase in average accuracy, 23–93%, when using between 1 and 10 training samples. As the number of samples increased from 10 to 40 a less than linear increase was achieved, leaving the authors to conclude that 15 samples at 93% recognition was a good configuration for detecting their gesture set.

Kela et al. [10] also used HMMs to recognize a set of 8 gestures. They studied gesture recognition accuracy in the context of a larger experiment comparing accelerometer-based gesture control to other control modalities. In the course of their investigation, they analyzed a set of 8 gestures and were able to achieve a mean recognition accuracy of 98.9% over all 8 gestures by using 12 training samples per gesture in a user dependent training configuration. Similarly, Park et al. [11] have examined gesture recognition accuracy in the context of energy efficient recognition techniques, working with hand-worn sensors and mobile devices. They examined a set of 8 gestures and collected gesture examples from 7 users in 4 different settings. The users performed each gesture while they were walking, running, standing or riding a car. For each gesture in each setting, each user provided 30 samples. Park et al. examined recognition accuracy in the context of energy effiency over this dataset. They investigated different configurations of HMMs on a Nokia N96 mobile phone, and in the best case, were able to achieve a mean recognition accuracy of 99.1% over all 8 gestures performed in a 'walking' setting, by using a 'mobility aware' HMM.

Pylvanainen applied HMMs to a slightly larger set of 10 gesture [12]. Both gesture dependent and independent training was conducted with data collected from a hand held mobile device containing accelerometers. The gesture dependent results suggested that only three training samples per gesture were needed to obtain 96.76% accuracy. While the gesture independent tests required a total of 21 (three samples from each of the seven participants) training samples per gesture to reach 99.76% accuracy. Pylvanainen does note that the higher accuracy seen in the user independent test was unexpected but representative of the minimal training samples used in the dependent experiment. Rehm used a Wiimote to recognize 3D gestures based on cultural specific interactions [13]. They conducted both user dependent and user independent experiments on a digit gesture set (10 gestures), a German emblem gesture set (7 gestures), and a VCR control gesture set (8 gestures) using Nearest Neighbor and Näive Bayes classifiers. For the digit gesture set, they claim accuracy as high as 100% for the Nearest Neighbor classifier in both the user dependent and independent cases and 58% accuracy for the Näive Bayes classifier in the user independent case. Accuracy for the German emblem gesture was 94% and 88% for Nearest Neighbor and Näive Bayes respectively in the user dependent case. For the VCR control gesture set, recognition accuracy was approximately 99% for both the Näive Bayes and Nearest Neighbor classifiers in the user dependent case. Mäntyjärvi also looked at 3D gesture recognition with 8 gestures using HMMs and found recognition accuracy in the high 90s as well [14].

Schlomer et al. [15] experimented with classifying 5 gestures performed with the Wiimote using HMMs. The gesture dependent results show that using 10 samples for training and the remaining five for recognition, resulted in classification accuracy between 85% and 95%. For the gestures Square, Circle and Z (also used in our study), the mean accuracy of 88.8%, 86.6%, and 94.3% were shown, respectively. Similarly, Kallio's collection of 16 gestures had 6 gestures in common with our set: Line to Right, Line to Left, Line Up, Line Down, Triangle, Parry (although Kallio mentions them by a different name). However, the 16 gestures can be broken up into four distinct gestures in four different orientations. The gestures are composed of time series data obtained from three acceleration sensors placed in a small wireless device. Then, by using HMMs, this work showed accuracy levels nearing 90%, with less than 10 training samples per gesture. However once the number of training samples was increased to 20, accuracy levels rose to over 95%. Kallio also discussed classification confusion between the Line to Right and Parry gestures when providing two samples for training [16].

Gestural text-entry systems have also been developed recently. Amma et al. [17] designed a custom 3D input device modeled on a glove, equipped with 3 accelerometers and 3 gyroscopes. They conducted a user study with 10 participants who provided 25 samples for each character of the english language. Each characted sample was segmented manually by pressing a button. Amma et al. investigated recognition accuracy for both individual characters and also for a small dictionary of english language words (consisting of 652 words), by using HMMs coupled with a language model. They were able to achieve a 95.3% accuracy over the set of english language characters, by using an HMM trained in user dependent configuration with 10 states and a mixture of 5 gaussians per state.