EG UK Computer Graphics & Visual Computing (2020)
P. D. Ritsos and K. Xu (Editors)

Gesture Recognition for Virtual Reality Motion Controllers

Christopher Headleand[1], Benjamin Williams[1], Jussi Holopainen[1], Marlon Gilliam[1],

[1]University of Lincoln, School of Computer Science, Lincoln, UK

Abstract

In this paper we discuss gesture recognition in the domain of Virtual Reality (VR) video games. We begin by presenting a detailed review of the literature. Furthermore, we discuss some of the specific opportunities and challenges that are specific to the VR domain. Most commercial VR devices come with tracked motion controllers as a default interface which facilitates the possibility of gesture control. However, video games specifically require a high degree of accuracy to prevent non-gesture actions being evaluated. To tackle this challenge we present a novel modification to the Hidden Markov Model gesture recognition approach. We expand on previous work with gestures in with the implementation of an adaptive database system allowing users to quickly engage with an application without significant training. Our results on a benchmark problem shows that the approach can produce impressive accuracy rates. The results from our benchmarking shows promise for the usability of gesture based interaction systems for VR devices in the future. Our system achieves high levels of recognition accuracy competitive with the best performing existing system whilst requiring minimal user independent training.

CCS Concepts
• ***Human-centered computing*** $\rightarrow$ ***Virtual reality;*** • ***Software and its engineering*** $\rightarrow$ ***Interactive games;***

1. Motivation

There is significant current research into how interaction in Virtual Reality can be improved and made more immersive. These approaches range from complex custom devices [CJT*10; HWH*19] to novel game experiences [AQSS16]. As the industry has moved towards motion controllers as a standard interface there is the possibility that these controllers could be used to capture gesture motions as an input mechanism.

A significant challenge in gesture recognition is that of user independent recognition [CAJ12; LZWV09] where different users can use the same device. Several studies have compared user dependent data and user independent data and results generally show a significant decrease in accuracy for independent recognition. More recent work [CAJ13; HYW*16] has achieved accuracies of around 90% comfortably. However, the methodologies for these studies have participants in an isolated environment when performing gestures. However, within the context of a game, the user's movements can be confused by other actions leading to significantly lower recognition accuracy.

Our motivation for this paper is to design an Adaptive Gesture Recognition system for motion control interfaces that can be used with a variety of consumer virtual reality devices. We will present a detailed review of the literature, before presenting a proposed solution.

2. Background

The *Nintendo Wii* brought motion controllers to commercial gaming and with it the possibility of accelerometer based gestures in games. One such example is Wiizards [KSL07] where users cast spells by performing gestures using the Wii controllers (Wiimotes) and a Hidden Markov Model (HMM) to classify the gesture. Their work builds on that of Payne et al. [PKE*06] who, prior to the Wii's release, developed a similar game with a simpler gesture set. Both works conclude that the use of gestures in games should be further explored.

Since the release of the Wii, other platforms have supported gesture control. For example, The Invoker [QS15] is a Role Playing Game (RPG) that uses the *Microsoft Kinect* for its control input.

2.1. Gesture Recognition in Virtual Reality Games

Various control methods have been applied to Virtual Reality throughout the history of the medium. Early application of gestures in VR was demonstrated by Weissmann and Salomon with their system using a data glove [WS99]. Their work achieved good recognition accuracy using a neural network to connect the data points on the glove and recognise several static hand poses.

Image processing has been a popular approach to gesture recognition in recent years. Cabral et al. [CMZ05] demonstrated the use of a rudimentary image processing gesture system to allow users to interact with a Powerwall. This work was further enhanced by Liu et al. [LYZ12] who developed a system that was capable of tracking and recognising 3 different hand gestures using image processing.

John et al. [JPD*17; HDR*16] present a VR wheelchair training environment. A Leap-Motion motion capture device is used to map the users hand's into the environment. This allows the user to perform gestures in the space and operate controls (such as elevator control panels).

There have also been games developed that have taken this approach. The experimental game 'Duke' combines the Oculus Rift with both the leap motion and the Kinect [AQSS16]. It leverages the Kinect for movement by tracking the user's legs and the leap motion for combat controls by tracking the user's hands. A work in progress game from Lee et al. [LPLK17] is developing a data glove that can be used in the place of the motion controllers for VR. They propose the use of gestures with the glove instead of button presses with a controller; the given example is clutching a sword by making a fist with the glove compared to pressing and holding a button on the controller.

3. Gesture Recognition Approaches

Outside of Virtual Reality, Gesture recognition has become a common mode of human-computer interaction largely due to developments in mobile computing and image processing technologies. Gesture recognition is changing to the way that we interact with many of our devices including smartphones, wearables, and gaming consoles [GP14]. Due to its widespread use, a large number of techniques have been developed that can be used in the classification and recognition of gestures.

In the following sub-sections we provide a brief review of the algorithms most commonly discussed in the literature. We identify from our research that a Hidden Markov Model (HMM) is the most appropriate algorithm for our recognition due to its reliability and accuracy with accelerometer systems as well as its proven use in games.

3.1. Dynamic Time Warping

The DTW algorithm works by calculating the distance between each possible pair of points between two signals and uses the distances to calculate a cumulative distance matrix [Sen08]. It then finds the least expensive path through this matrix. It usually features normalised signals and only one-dimensional series [RA15]. DTW techniques can achieve high accuracy results [AV10; LZWV09] with minimal training sets but their accuracies are affected by amplitude variation [HYW*16] which is often an issue when dealing with accelerometer data due to tilt [AV10].

Improvements have been made to the accuracies of DTW in recognition on Kinect data by using feature weighting [RDE11] which was further explored and improved by [CATA13]. Lu et al. [LCL*14] show a DTW algorithm achieving working with a set of 19 gestures. The system does have lower accuracies than others in the field but its fast computation time on a low power device is impressive considering the larger gesture set.

3.2. Support Vector Machines

A Support Vector Machine (SVM) works by separating the data classes into hyperplanes in a high dimensional space. It then finds classifications from unseen instances in the hyperplanes [Bur98]. SVMs are known to provide great classification performance [WPZQ09] with low run time computational costs [HYW*16]. However SVMs are usually used for static gesture recognition and also require large amounts of training time [RA15] with large data-sets containing high class feature counts to gain greater accuracy [HYW*16].

Wu considers the performance of an SVM using temporal data with an accelerometer and shows it achieving high accuracy rates in user dependent and independent cases [WPZQ09], however the system uses data sets in the thousands for training. He uses *feature fusion* to achieve high accuracy results with mobile accelerometers [He11].

3.3. Neural Networks

Time delay neural networks (TDNNs) have been widely used for the detection of 2D motion trajectories [WA99; YAT02] as well as the segmentation of video data based on skin colour recognition [SSA03]. TDNNs are multi-layer networks that slide over small windows of data in a temporal domain making local decisions; this allows for them to be used in continuous recognition. [PS15; RA15].

Multi-Layer Perceptron (MLP) [RHW88] models can be used to distinguish between non-linearly separable data [APF12]. They are also used in the recognition of accelerometer based gestures for culturally specific interactions [RBA08] where they achieve comparable accuracy results in user dependent cases.

3.4. Linear Classifiers

The Adaboost algorithm is a machine learning meta-algorithm [FS95] that is used as the base recognition system in the Kinect [Mic14]. The Adaboost algorithm can be used to combine multiple classifiers in order to create a blended classifier that can achieve impressive results [CGP07]. However in comparison to other recognition methods it has been seen to be less accurate [HVL10]. Hoffman et al. go on to discuss how this result was unexpected due to the sophistication of the classifier and show a linear classifier based on Rubine's method [Rub91] outperforming the Adaboost classifier. Lu et al. [LCL*14] also demonstrated how a Bayes linear classifier can be used in the calculation of smaller gestures on a mobile accelerometer device.

3.5. Hidden-Markov-Models

Hidden Markov models are doubly stochastic processes that are governed by an underlying Markov chain with a finite number of states where each state is associated with a set of random functions. The model represents a number of states that move through series of time steps. Each state can be described by its transition probability and its output probability. At each time step the sequence can either output the value from its alphabet associated with the current state or transition to the next state. Previous to its role in gesture recognition the HMM was applied to speech recognition [Rab89].

It was then introduced as a method for gesture recognition [YX94] where it was used to recognise the numbers 0 to 9, drawn in 2D with mouse input, with remarkable accuracies of up to 99.78%. Following this, vision based gesture recognition was achieved on the American Sign Language [SP96]. They noted some important findings in that often with continuous detection, gestures would frequently double up if the path was of a certain manner. Numbers and letters from the English sign language were recognised in [RAM09] using a HMM to recognise trajectory data and an SVM for the recognition of hand postures. Their high accuracies highlight the strengths of both recognition methods and give a good example of how they can be used in tandem. Premaratne et al. [PYVI17] recognise 16 letters from the English Language. They implement an initial segmentation using direction from origin to categorise gestures into 5 groups based off the stroke directions of: up, down, left, right, and general diagonal. Their system achieves good accuracy and provides the benefit of lowering computation times.

In 2004 the XWand [WW04] introduced the idea of physical UI devices being used for gesture recognition using data from accelerometers. The XWand was tested using: a Linear Time warping recognition method; a Dynamic Time Warping method and a HMM system. The HMM approach outperformed the other two systems by a significant factor. Following this further proof of HMM working with accelerometer data was presented in both [MKKK04] and [KKM*06]. They introduced a method of training using noise models to produce variations of a single gesture sequence that could then build a user dependent database. Mantyjarvi et al. [MKKK04] prove that when using vector quantization on the accelerometer data with the k-means algorithm a value of k being 8 is best for 2D data. Kela et al. [KKM*06] showed that there was user interest in gesture control for home electronic devices and established a competent 3D accelerometer system to recognise them.

HMMs were proven to work successfully with the user dependent accelerometer data recorded from the Nintendo Wiimote [08] with user trained data sets. Schlomer et al. [08] follows Mantyjarvi's work on quantization finding k=14 to be suitable for 3D accelerometer data. Further work [CAJ13] achieves near perfect accuracy results for explicit recognition of gestures in user dependent and independent testing, once again using a Wiimote. Their work consists of a larger degree of gestures. Criticisms have however been made to their findings as they only use gestures "confined to a 2D planar motion" [Ars14]. Where Arsenault's [Ars14] findings show strong accuracy results for his self defined complex gestures, the set size is much lower than both the gesture sets used by Chen et al. [CAJ13]. In recent work Whitehead [Whi18] demonstrates significant accuracy on a wearable accelerometer for a game application. They translate the 3D accelerometer data into a single data string using a cubic bisection method to outline positions. They consider how users have a desire to improve their use of the system therefore wanting to learn the gestures and how this matches a game environment where users generally strive for improvement. Interestingly for the context of game use they notice that often the accuracy rates are highly user dependent depending on the skill and memory of the user.

3.6. Continuous Recognition

The majority of previously discussed gesture recognition systems have the user dictate when a gesture starts and sometimes when it ends as well. This allows for easy segmentation and extraction of the gesture data but has been said to be less natural and user friendly [RLNS10].

In early works Jung-Bae et al. present a system that is capable of continuously recognising 18 words in Korean sign language sentences making use of a HMM for classification [KPBB02]. They use changes in angle of trajectory to separate the data into different HMMs allowing for complex symbols to be recognised using simple individual gestures. They then use a difference in the speed of the hand to detect the start and end of words. The *Zero-Codeword Detection System* uses a similar system to separate the recognition of different numbers [EAAM08]. The system uses a minimum number of pixels per frame to calculate the static velocity and uses a minimum threshold on this value to determine the start and end of gestures.

Raffa et al. [RLNS10] show a system that is capable of achieving continuous accuracy results that are equal to explicit ones. They achieve this on low budget hardware using accelerometers and discuss how it has the ability to save battery life on devices using their HMM garbage model which works by discarding many non valid gestures before they reach the intensive computational stage of recognition within the HMM.

Several recent works explore the possibility of using neural networks to tackle continuous recognition of the *ChaLearn* [WLZ*16] data set which contains 47933 examples of 249 different gestures [WSHC16; CHKB17; CLY*17]. While the accuracies of these works are not in the same range as many smaller scale recognition systems they provide strong grounding for how large scale continuous recognition could be solved in the future.

4. Adaptive Database HMM

It has been considered previously that gesture recognition systems should be "flexible and expandable in order to maximise efficiency, accuracy and understandability." [RA15]. With this in mind we propose a method that adapts an independent database as users interact with the system. This approach was first theorised by Kratz et al [KSL07]. In this section we detail the program design and implementation of our gesture recognition system.

4.1. Motion Tracking

We are using the touch controllers that come with the Oculus Rift headset. These controllers are intuitive, and allow for micro-gestures (such as opening and closing of a hand) which are commonly used in games for actions including user locomotion [AH18]. The touch controllers give us access to a large amount of motion data. Unlike many present accelerometer systems or conventional 2D image processing the rift motion controllers are closer

Figure 1: *The Oculus Rift and wireless motion controllers in use.*

to a depth camera system like the Kinect (see figure 4.1). However, while the Kinect has data for everything in frame we are limited to the three points of motion, namely the head and hands of our users. Due to the direct mapping of movements into 3D space we can extrapolate position and orientation from which we can derive acceleration and velocity for both.

To track motion we attach a listener to both touch controllers, which mimic all their transformations. These listeners are attached to the players headset as a child object meaning that their positions are relative to the headset's current position and orientation. To extrapolate the data from the listeners we poll their transform data during the fixed update loop within unity. The position and rotation are both extracted as vectors using their Euler values for rotation. These values are passed through to the recogniser and stored in lists. When being parsed for either storage or recognition these values are separated into arrays of double values so that they can be passed into the HMM.

4.2. Two-handed Classification

For the purpose of testing we will only consider gestures that use one hand. We hope that in only using one hand to record the gestures it prevents users from being overwhelmed by the gestures. However, it is worth noting that in theory both hands could be used in the system.

However, we do consider the use for two handed gestures in the future. The system is designed in such a way that the number of inputs could simply be increased to include another hand. An alternative method could be the use of another system in parallel that performs individual classification on each hand. There is much room for exploration here with varying thresholds and system configurations. As for teaching users to use the system the application could be structured such that users are first introduced to single handed gestures and then later shown two handed gestures. The two handed gestures could also be designed in a fashion where their uses are related to the single handed gestures that they incorporate so as to keep the motions related for the user.

4.3. HMM Topology

The topology of a HMM consists of several variables that are used to determine the way in which the HMM interpolates data, along with the algorithms that the HMM uses in the training and classification of said data. For our system we use the HMM from the *Accord.NET* machine learning library as it is written primarily in C♯ and therefore can be easily integrated into Unity3d projects.

Motion gestures are "order-constrained time-evolving" signals therefore the left-right HMM topology is most often considered to be the most appropriate [MKKK04; CAJ13]. Although there have been publications that argue that there is no difference between ergodic and left-right topologies [HHH98; 08] we will be using left-right as it stands as the most commonly used in modern HMM gesture recognition systems. The Baum-Welch algorithm is used to train model parameters and the forward algorithm is used in the probabilistic evaluations [BP66]. Schlomer et al. find that the number of states in the HMM has very little impact on the accuracy of the model after a certain threshold [08]. We confirm this in our testing and settle on 7 states. No minimum matching threshold is applied to the classification of gestures.

4.4. Database Adaption

One of the challenges that we aim to address in this work is that of improving user independent recognition rates. Our proposed method makes use of database adaption where the original user independent data is subsidised by user data throughout the game. This solution was found after considering proposed methods from [KSL07]. There are several approaches that can be adopted these include expanding the database, overwriting the database, overwriting and then shrinking the database, and expanding and shrinking the database. The implementation of these systems are relatively simple. When a new gesture is recorded we either add it to the database or we find an element with the same key as it and then remove that element before adding the new one.

One decision that must be made here is when to update the database. Simply adding a gesture when it comes out of the HMM runs the risk of adding a false positive to the database. This can be countered by having a form of user validation where they must confirm that the gesture recognised was correct.

5. Method

In the body of literature there are a number of methods used to evaluate gesture recognition systems. The most commonly adopted

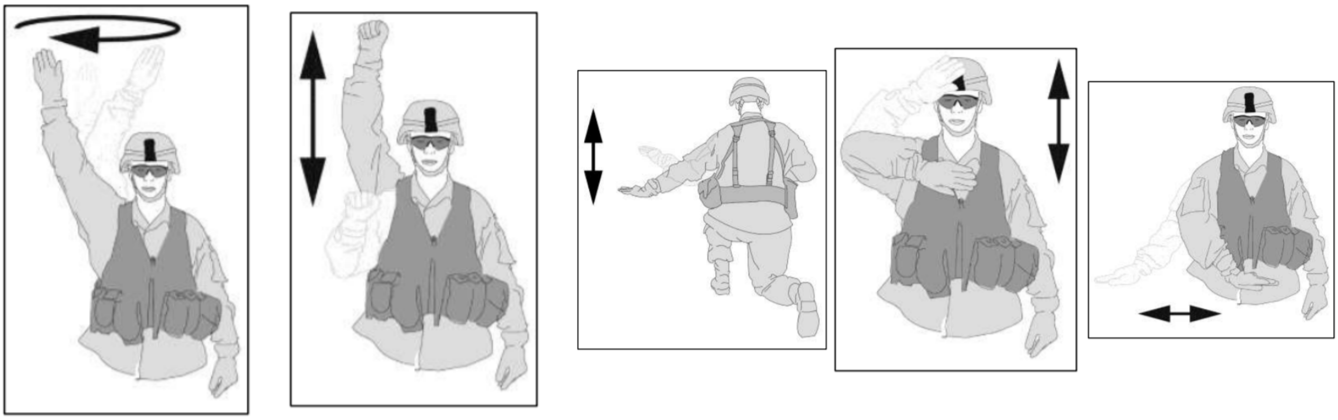

Figure 2: *The military command gestures [***militarygestures***]. From left to right, follow, rush, sneak, cease firing, start and firing.*

approach involves taking average accuracy results across all gestures, using a testing dataset. In this approache a large dataset is collected, and split, with half being used for training, and the other half used for testing.

We decided to use two gesture sets to evaluate the Adaptive Database HMM. The first dataset we used was numbers from 0 to 9. Numbers are commonly used in gesture recognition testing as they are distinct patters that users will be familiar with. We use this as our first benchmark to allow our work to be better compared with the body of literature.

The second gesture set that we chose to use were military gestures. The long-term goal of this project is to implement our system into a first-person shooter video game. For this reason, we decided to use military gestures as they can be directly applied to this use-case. Specifically, we want the player to be able to communicate with and command other characters in the game using gestures. Another benefit of using military gestures is that their usability and clarity has been extensively tested in high-pressure environments. The selected gestures from the military database are detailed in figure .1.

6. System Testing and Results

Following the completion of the core of the system we begin batch testing with a number of independent variables. The variables we consider are the number of points trained along each gesture and the number of instances of each gesture in the database.

For this test, we used the numbers from 0 to 9 to test our system. Numbers are commonly used characters in gesture recognition, so they made a good candidate for evaluation. For this initial testing an individual who was comfortable with motion in VR recorded a set of 50 instances for each number from 0 to 9. These gestures were then split between a training set, and a testing set. The results of this data are seen in table 1. For each number 25 gestures were tested against the HMM leaving a possible total of 250 gestures to recognise.

The results of the initial testing show that the recognition system

Table 1: *Accuracy rates for user dependent database. Top axis defines number of states in the HMM and left axis defines the size of the training database.*

	1	2	3	4	5	6	7
1	36%	46%	54%	56%	54%	60%	58%
2	44%	50%	64%	66%	70%	68%	70%
3	48%	54%	64.8%	68%	74%	76.4%	76%
4	50%	54.8%	66%	72%	76.4%	80%	80.8%
5	58%	64%	72%	78.4%	82.8%	86.4%	88%
6	64%	68.4%	74.8%	80.4%	83.6%	88.4%	90%
7	66%	69.2%	75.6%	82.8%	88.8%	89.6%	92%
8	66.4%	71.2%	76.4%	83.6%	88.8%	91.2%	94.4%
9	66.4%	72.4%	77.2%	85.6%	89.6%	92.4%	95.6%
10	66.8%	73.6%	80.8%	88%	90.4%	94.4%	96.4%
11	67.2%	75.2%	81.6%	89.2%	90.8%	94.8%	96.8%
12	66.8%	76.4%	83.2%	92%	93.2%	95.2%	97.2%
13	66.4%	77.2%	83.6%	93.6%	92.4%	95.6%	97.6%
14	67.2%	77.6%	83.6%	93.6%	93.6%	95.6%	97.6%
15	67.6%	77.6%	85.6%	93.6%	93.6%	96.4%	98%
16	66.8%	78%	86.8%	94.4%	94.4%	98%	98.4%
17	68%	77.6%	88%	95.2%	95.6%	98.4%	98.4%
18	67.6%	78.4%	88.4%	96.4%	97.2%	98.4%	98.4%
19	67.2%	78.4%	88.8%	96.8%	98%	99.2%	98.8%
20	67.6%	78.8%	88.8%	96.8%	98.4%	99.2%	99.2%

performs very well with 7 states. The states show to have a significant impact at the lower numbers which we believe is due to the similarity of the path data of several of the gestures. Since all the numbers were drawn in roughly the same z plane with a low number of state it is likely that many number shared the same location for some states. This matches the findings from our later results when testing military gestures in table 2.

We also find from these results that after a database size of 10 the increase in accuracy becomes very steady with small improvements coming from every few extra gestures in the set. This allows us to consider a database size of around 10 for use in the studies which allows for very fast training times and means that modifications made using the adaptive database could have a larger impact.

6.1. Testing Military Gestures

To test the military command gestures we followed the same testing methodology as the numbers, notably creating a data-set and splitting this into a training set and a testing set.

One notable thing that can be seen from the initial testing is that accuracy results in Table 2 with very few states can manage to achieve relatively strong accuracy. We believe this to be due to the fact that our gestures all take place in fairly distinct positions therefore meaning that segmentation based on one step gives a strong indication of the gesture that is about to be performed. Thus, with relatively few steps we can can observe moderately distinct results; especially within only a single user test environment. This is consistent through all database sizes and even sees usable accuracy results with only one state. We do not consider this the best for practical use as once user independent use is introduced this may begin to vary. We can see that after size 6 the accuracy results seem to stabilise over 90% and after size 10 they begin to incrementally increase at a generally stable rate of 0.4% (or one gesture) for every subsequent size increase.

The main difference in these accuracies when compared with previous tests using the numbers is that they achieve much higher recognition rates at the lower level. This demonstrates how systems can be optimised by using gestures that are well designed to be used together. The key here is that each gesture should aim to be different from the others in the metric that is being used for motion detection. For ours it is having them in differing positions but using an accelerometer based system this could be related to both speed and motion of the gesture.

Table 2: *Accuracy rates for initial single user database.*

	1	2	3	4	5	6	7
1	82	72	54	54	54	52	48
2	80	84	86	84	80	80	80
3	86	84	84	84	86	86	86
4	82	82	84	84	88	90	90
5	88	90	92	92	92	92	94
6	92	92	92	94	94	94	94
7	93.6	94	94	94	94	94	94
8	94	94	94.4	94.4	94	94.4	94.4
9	94.4	94.4	94.4	95.6	95.6	95.6	95.6
10	96	96.4	96	96	96.4	96.4	96.4
11	96.4	96.4	96.8	96.8	96.8	96.8	96.8
12	97.2	97.2	97.2	97.2	97.2	97.2	97.2
13	97.2	97.2	97.6	97.6	97.6	97.6	97.6
14	97.2	97.2	97.6	97.6	97.6	97.6	97.6
15	97.6	97.6	97.6	97.6	97.6	97.6	98
16	98	98	98	98	98	98	98.4
17	98	98	98	98	98	98.4	98.4
18	98.4	98.4	98.4	98.4	98.4	98.4	98.4
19	98.4	98.4	98.8	98.8	98.8	99.2	98.8
20	98.8	98.8	98.8	98.8	99.2	99.2	99.2

7. Conclusion

This paper presents a gesture recognition system designed to work with modern VR devices whilst also being flexible in its design to allow for development in the future. We discuss the approach of combining a Hidden Markov Model (HMM) recognition algorithm with an adaptive database to allow it to be biased towards specific users. Allowing a user-independent system to be made semi user-dependent over time. This allows for higher accuracies with reduced training. We conclude by testing our system against two benchmarks, notably number gestures (a common benchmark) and a military gesture benchmark (chosen due to its applicability to video games and training environments). Our system performs admirably demonstrating its versatility and accuracy.

7.1. Future Work

Following benchmarking we now plan to apply the system in a video game and evaluate the user experience. This is the long-term goal of this project, to produce a system that can allow a user to interact naturally with artificial intelligence characters in a VR game.

Furthermore, we have developed a continuous recognition version of this system that addresses an entirely new method of control input for VR devices. The hope is that the continuous recognition approach could allow users to seamlessly interact with a game, without the need to break their suspension of disbelief to explicitly define a gesture.

References

[08] "Gesture recognition with a Wii controller". New York, New York, USA: ACM Press, 2008, 11 3, 4.

[AH18] AP CENYDD, LLYR and HEADLEAND, CHRISTOPHER J. "Movement modalities in virtual reality: A case study from Ocean Rift examining the best practices in accessibility, comfort, and immersion". *IEEE Consumer Electronics Magazine* 8.1 (2018), 30–35 3.

[APF12] ANJO, M. D. S., PIZZOLATO, E. B., and FEUERSTACK, SEBASTIAN. "A real-time system to recognize static gestures of Brazilian sign language (libras) alphabet using Kinect". *Proceedings of the 11th Brazilian Symposium on Human Factors in Computing Systems - IHC '12* 5138 (2012), 259–268 2.

[AQSS16] AMIR, MOHD HEZRI, QUEK, ALBERT, SULAIMAN, NUR RASYID BIN, and SEE, JOHN. "DUKE". *Proceedings of the 13th International Conference on Advances in Computer Entertainment Technology - ACE2016*. New York, New York, USA: ACM Press, 2016, 1–6 1, 2.

[Ars14] ARSENAULT, DENNIS. "A quaternion-based motion tracking and gesture recognition system using wireless inertial sensors". PhD thesis. 2014 3.

[AV10] AKL, AHMAD and VALAEE, SHAHROKH. "Accelerometer-based gesture recognition via dynamic-time warping, affinity propagation, & compressive sensing". *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* April (2010), 2270–2273 2.

[BP66] BAUM, LEONARD E. and PETRIE, TED. "Statistical Inference for Probabilistic Functions of Finite State Markov Chains". *The Annals of Mathematical Statistics* 37.6 (1966), 1554–1563 4.

[Bur98] BURGES, CHRISTOPHER J C. "A Tutorial on Support Vector Machines for Pattern Recognition". *Data Mining and Knowledge Discovery* 2 (1998), 121–167 2.

[CAJ12] CHEN, MINGYU, ALREGIB, GHASSAN, and JUANG, BIING HWANG. "A new 6D motion gesture database and the benchmark results of feature-based statistical recognition". *2012 IEEE International Conference on Emerging Signal Processing Applications, ESPA 2012 - Proceedings* (2012), 131–134 1.

[CAJ13] CHEN, MINGYU, ALREGIB, GHASSAN, and JUANG, BIING HWANG. "Feature processing and modeling for 6D motion gesture recognition". *IEEE Transactions on Multimedia* 15.3 (2013), 561–571 1, 3, 4.

[CATA13] CELEBI, SAIT, AYDIN, ALI S, TEMIZ, TALHA T, and ARICI, TARIK. "Gesture Recognition using Skeleton Data with Weighted Dynamic Time Warping". *Proceedings of the International Conference on Computer Vision Theory and Applications*. SciTePress - Science, 2013, 620–625 2.

[CGP07] CHEN, QING, GEORGANAS, NICOLAS D, and PETRIU, EMIL M. "Real-time Vision-based Hand Gesture Recognition Using Haar-like Features". *2007 IEEE Instrumentation & Measurement Technology Conference IMTC 2007* (2007), 1–6 2.

[CHKB17] CAMGOZ, NECATI CIHAN, HADFIELD, SIMON, KOLLER, OSCAR, and BOWDEN, RICHARD. "Using Convolutional 3D Neural Networks for User-independent continuous gesture recognition". *Proceedings - International Conference on Pattern Recognition*. 2017, 49–54 3.

[CJT*10] CONNELLY, LAURI, JIA, YICHENG, TORO, MARIA L., et al. "A pneumatic glove and immersive virtual reality environment for hand rehabilitative training after stroke". *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 18.5 (2010), 551–559 1.

[CLY*17] CHAI, XIUJUAN, LIU, ZHIPENG, YIN, FANG, et al. "Two streams Recurrent Neural Networks for Large-Scale Continuous Gesture Recognition". *Proceedings - International Conference on Pattern Recognition*. 2017, 31–36 3.

[CMZ05] CABRAL, MARCIO C., MORIMOTO, CARLOS H., and ZUFFO, MARCELO K. "On the usability of gesture interfaces in virtual reality environments". *Proceedings of the 2005 Latin American conference on Human-computer interaction - CLIHC '05* OCTOBER 2005 (2005), 100–108 1.

[EAAM08] ELMEZAIN, MAHMOUD, AL-HAMADI, AYOUB, APPENRODT, JORG, and MICHAELIS, BERND. "A Hidden Markov Model-based continuous gesture recognition system for hand motion trajectory". *2008 19th International Conference on Pattern Recognition* January (2008), 1–4 3.

[FS95] FREUND, YOAV and SCHAPIRE, ROBERT E. "A desicion-theoretic generalization of on-line learning and an application to boosting". 1995, 23–37 2.

[GP14] GILLIAN, NICHOLAS and PARADISO, JOSEPH A. "The Gesture Recognition Toolkit". *Journal of Machine Learning Research* 15.November (2014), 3483–3487 2.

[HDR*16] HEADLEAND, CHRISTOPHER J, DAY, THOMAS W, R, SERBAN, et al. "A cost-effective virtual environment for simulating and training powered wheelchairs manoeuvres". *Proc. Med. Meets Virtual Reality NextMed/MMV* 220 (2016), 134 2.

[He11] HE, ZHENYU. "Accelerometer based gesture recognition using fusion features and SVM". *Journal of Software* 6.6 (2011), 1042–1049 2.

[HHH98] HOFMANN, FRANK G., HEYER, PETER, and HOMMEL, GÜNTER. "Velocity profile based recognition of dynamic gestures with discrete Hidden Markov Models". 1998, 81–95 4.

[HVL10] HOFFMAN, MICHAEL, VARCHOLIK, PAUL, and LAVIOLA, JOSEPH J. "Breaking the status quo: Improving 3D gesture recognition with spatially convenient input devices". *2010 IEEE Virtual Reality Conference (VR)*. IEEE, 2010, 59–66 2.

[HWH*19] HARRINGTON, JAKE, WILLIAMS, BEN, HEADLEAND, CHRISTOPHER, et al. "A Somatic Approach to Combating Cybersickness Utilising Airflow Feedback". (2019) 1.

[HYW*16] HONG, FENG, YOU, SHUJUAN, WEI, MEIYU, et al. "MGRA: Motion gesture recognition via accelerometer". *Sensors (Switzerland)* 16.4 (2016), 1–25 1, 2.

[JPD*17] JOHN, NIGEL W, POP, SERBAN R, DAY, THOMAS W, et al. "The implementation and validation of a virtual environment for training powered wheelchair manoeuvres". *IEEE transactions on visualization and computer graphics* 24.5 (2017), 1867–1878 2.

[KKM*06] KELA, JUHA, KORPIPÄÄ, PANU, MÄNTYJÄRVI, JANI, et al. "Accelerometer-based gesture control for a design environment". *Personal and Ubiquitous Computing* 10.5 (2006), 285–299 3.

[KPBB02] KIM, JUNG-BAE, PARK, KWANG-HYUN, BANG, WON-CHUL, and BIEN, Z ZENN. "Continuous gesture recognition system for Korean sign language based on fuzzy logic and hidden Markov model". *2002 IEEE World Congress on Computational Intelligence. 2002 IEEE International Conference on Fuzzy Systems. FUZZ-IEEE'02. Proceedings (Cat. No. 02CH37291)*. Vol. 2. IEEE. 2002, 1574–1579 3.

[KSL07] KRATZ, LOUIS, SMITH, MATTHEW, and LEE, FRANK J. "Wiizards: 3D gesture recognition for game play input". *Proceedings of the 2007 conference on Future Play Future Play 07* (2007), 209–212 1, 3, 4.

[LCL*14] LU, ZHIYUAN, CHEN, XIANG, LI, QIANG, et al. "A hand gesture recognition framework and wearable gesture-based interaction prototype for mobile devices". *IEEE Transactions on Human-Machine Systems* 44.2 (2014), 293–299 2.

[LPLK17] LEE, SEOKWON, PARK, KIHONG, LEE, JUNYEOP, and KIM, KIBUM. "User Study of VR Basic Controller and Data Glove as Hand Gesture Inputs in VR Games". *Proceedings - 2017 International Symposium on Ubiquitous Virtual Reality, ISUVR 2017* (2017), 1–3 2.

[LYZ12] LIU, YUN, YIN, YANMIN, and ZHANG, SHUJUN. "Hand Gesture Recognition Based on HU Moments in Interaction of Virtual Reality". *2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics*. IEEE, 2012, 145–148 2.

[LZWV09] LIU, JIAYANG, ZHONG, LIN, WICKRAMASURIYA, JEHAN, and VASUDEVAN, VENU. "uWave: Accelerometer-based personalized gesture recognition and its applications". *Pervasive and Mobile Computing* 5.6 (2009), 657–675 1, 2.

[Mic14] MICROSOFT. *Kinect gesture builder*. 2014. URL: https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785529(v=ieb.10) 2.

[MKKK04] MÄNTYJÄRVI, JANI, KELA, JUHA, KORPIPÄÄ, PANU, and KALLIO, SANNA. "Enabling fast and effortless customisation in accelerometer based gesture interaction". *Proceedings of the 3rd international conference on Mobile and ubiquitous multimedia*. 2004, 25–31 3, 4.

[PKE*06] PAYNE, JOHN, KEIR, PAUL, ELGOYHEN, JOCELYN, et al. "Gameplay issues in the design of spatial 3D gestures for video games." *CHI '06 extended abstracts on Human factors in computing systems - CHI EA '06* (2006), 1217 1.

[PS15] PISHARADY, PRAMOD KUMAR and SAERBECK, MARTIN. "Recent methods and databases in vision-based hand gesture recognition: A review". *Computer Vision and Image Understanding* 141 (2015), 152–165 2.

[PYVI17] PREMARATNE, PRASHAN, YANG, SHUAI, VIAL, PETER, and IFTHIKAR, ZUBAIR. "Centroid tracking based dynamic hand gesture recognition using discrete Hidden Markov Models". *Neurocomputing* 228.February 2016 (2017), 79–83 3.

[QS15] QUEK, ALBERT and SEE, JOHN. "The invoker: Intuitive gesture mechanics for motion-based shooter RPG". *2015 Game Physics and Mechanics International Conference, GAMEPEC 2015 - Proceeding* (2015), 6–10 1.

[RA15] RAUTARAY, SIDDHARTH S and AGRAWAL, ANUPAM. "Vision based hand gesture recognition for human computer interaction: a survey". *Artificial intelligence review* 43.1 (2015), 1–54 2, 3.

[Rab89] RABINER, L.R. "A tutorial on hidden Markov models and selected applications in speech recognition". *Proceedings of the IEEE* 77.2 (1989), 257–286 3.

[RAM09] RASHID, OMER, AL-HAMADI, AYOUB, and MICHAELIS, BERND. "A framework for the integration of gesture and posture recognition using HMM and SVM". *Proceedings - 2009 IEEE International Conference on Intelligent Computing and Intelligent Systems, ICIS 2009* 4 (2009), 572–577 3.

[RBA08] REHM, MATTHIAS, BEE, NIKOLAUS, and ANDRÉ, ELISABETH. "Wave like an Egyptian—accelerometer based gesture recognition for culture specific interactions". *People and Computers XXII Culture, Creativity, Interaction 22* (2008), 13–22 2.

[RDE11] REYES, MIGUEL, DOMINGUEZ, GABRIEL, and ESCALERA, SERGIO. "Featureweighting in dynamic timewarping for gesture recognition in depth data". *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*. IEEE, 2011, 1182–1188 2.

[RHW88] RUMELHART, D.E., HINTON, G.E., and WILLIAMS, R.J. "Learning Internal Representations by Error Propagation". *Readings in Cognitive Science*. Elsevier, 1988, 399–421 2.

[RLNS10] RAFFA, GIUSEPPE, LEE, JINWON, NACHMAN, LAMA, and SONG, JUNEHWA. "Don't slow me down: Bringing energy efficiency to continuous gesture recognition". *Proceedings - International Symposium on Wearable Computers, ISWC*. 2010 3.

[Rub91] RUBINE, DEAN. "Specifying gestures by example". *ACM SIGGRAPH Computer Graphics* 25.4 (July 1991), 329–337 2.

[Sen08] SENIN, PAVEL. "Dynamic Time Warping Algorithm Review". *Science* 2007.December (2008), 1–23 2.

[SP96] STARNER, THAD and PENTLAND, ALEX SANDY. "Real-Time American Sign Language Recognition Hidden Markov Models from Video Using". *AAAI Technical Report FS-96-05* (1996), 109–116 3.

[SSA03] SIGAL, LEONID, SCLAROFF, STAN, and ATHITSOS, VASSILIS. "Skin Color-Based Video Segmentation under Time-Varying Illumination". (2003) 2.

[WA99] WOHLER, CHRISTIAN and ANLAUF, JOACHIM K. "An adaptable time-delay neural-network algorithm for image sequence analysis". *IEEE Transactions on Neural Networks* 10.6 (1999), 1531–1536 2.

[Whi18] WHITEHEAD, ANTHONY D. "Gesture Recognition with Accelerometers for Game Controllers, Phones and Wearables". *GSTF Journal on Computing (JoC)* 3.4 (2018) 3.

[WLZ*16] WAN, JUN, LI, STAN Z, ZHAO, YIBING, et al. "ChaLearn Looking at People RGB-D Isolated and Continuous Datasets for Gesture Recognition". *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. 2016, 761–769 3.

[WPZQ09] WU, JIAHUI, PAN, GANG, ZHANG, DAQING, and QI, GUANDE. "Gesture recognition with a 3-d accelerometer". *Proceedings of the 6th International Conference on Ubiquitous Intelligence and Computing* 5585 (2009), 25–38 2.

[WS99] WEISSMANN, J. and SALOMON, R. "Gesture recognition for virtual reality applications using data gloves and neural networks". *Neural Networks, 1999. IJCNN'99.* 3 (1999), 2043–2046 1.

[WSHC16] WANG, PEISONG, SONG, QIANG, HAN, HUA, and CHENG, JIAN. "Sequentially Supervised Long Short-Term Memory for Gesture Recognition". *Cognitive Computation* 8.5 (2016), 982–991 3.

[WW04] WILSON, DANIEL and WILSON, ANDY. "Gesture Recognition Using The XWand". (2004) 3.

[YAT02] YANG, MING-HSUAN, AHUJA, NARENDRA, and TABB, MARK. "Extraction of 2d motion trajectories and its application to hand gesture recognition". *IEEE Transactions on pattern analysis and machine intelligence* 24.8 (2002), 1061–1074 2.

[YX94] YANG, JIE and XU, YANGSHENG. *AD-A282 845 Hidden Markov Model for Gesture Recognition*. Tech. rep. 1994 3.