Caserman, P., Tregel, T., Fendrich, M., Kolvenbach, M., Stabel, M., & Göbel, S. (2018, November). Recognition of Full-Body Movements in VR-Based Exergames Using Hidden Markov Models. In Joint International Conference on Serious Games (pp. 191-203). Springer, Cham

Recognition of Full-Body Movements in VR-based Exergames using Hidden Markov Model

Polona Caserman[1], Thomas Tregel[1], Marco Fendrich[2], Moritz Kolvenbach[2], Markus Stabel[2], and Stefan Göbel[1]

Multimedia Communications Lab - KOM, Technische Universität Darmstadt, Darmstadt, Germany
[1] {firstname.lastname}@kom.tu-darmstadt.de
[2] {firstname.lastname}@stud.tu-darmstadt.de

Abstract. Due to recent improvements in Virtual Reality (VR) regarding the potential of full-body tracking, the number of VR-based exergames has been increasing. However, such applications often depend on additional tracking technology, e.g., markerless or marker-based. On the one hand, tracking approaches, such as the Kinect device are limited by either high latency or insufficient accuracy. On the other hand, motion capture suits are expensive and create discomfort. In this paper we present an accurate motion recognition approach, using only the HTC Vive HMD with their associated Controllers and Trackers. The recognition is based on an Hidden Markov Model, that has been trained in advance for a specific movement. The results suggest that our system is capable of detecting a complex full-body gesture, such as yoga *Warrior I* pose, with an accuracy of 88%. In addition, audible feedback is provided, so that the user can immediately hear if the particular exercise has been executed correctly. Such a system can be used to assist players in learning a particular movement and can be applied in various serious games applications, e.g., for training purposes or rehabilitation.

Keywords:

1 Introduction

Recognition of body movements is attracting increasing research interest. However, most publications using Hidden Markov Models (HMMs) are recognizing only a part of the body, e.g., hand gestures [4, 12–14, 18, 24] or arm movements [17,20]. Only a little research regarding full-body recognition in VR-based exergames using machine learning approaches has been made.

On the one hand, many publications trying to recognize full-body motions are using markerless tracking technologies, such as the Microsoft Kinect device[1] which provides the joint orientation for the player's skeleton. However, the

[1] https://www.xbox.com/en-GB/xbox-one/accessories/kinect, last visited on May 25th, 2018

The final authenticated version is available online at https://doi.org/10.1007/978-3-030-02762-9_20.

Kinect suffers from occlusion, low precision, and inaccuracy [8]. On the other hand, more precise motion capture technologies, such as Optitrack[2] or Vicon[3], are based on multiple markers attached to the player's suit which are tracked by multiple cameras. Even though such a system is more accurate, it can create discomfort since the user has to wear a tight suit [21]. Moreover, multiple Inertial Measurement Units (IMUs) can be attached to the user's body in order to recognize body motions [7]. However, since multiple sensors must be worn by the user, such a system may also be considered intrusive.

In contrast to Kinect, our approach is designed to recognize full-body motions reliably. Furthermore, the recognition algorithm should be independent of the execution speed (i.e. how fast or slow the user performs a pose) as well as user's size (i.e. the height of the user). According to Raheja et al. [20], HMM are scale-invariant. In other words, the recognition algorithm using HMM works regardless of the student's height performing a movement. Unlike suit-based motion capture systems, we use only a small number of devices. Some advantages of the reduced sensor amount include ease of use, detection reliability and less potential for error. With two HTC Vive Controllers, the hands' movements are tracked. With the three additional HTC Vive Trackers, we determine the position of the lower back as well as both legs.

There are basically three main criteria that our approach should meet:

- The approach must be computationally efficient and must recognize the motion in real-time
- The motion recognition algorithm must be reliable:
  - It should be independent of the player's movement speed
  - It should provide similar results regardless of the player's height
- Only a small number of devices must be used

The paper is organized as follows: Section 2 presents relevant related work. In Section 3, the concept of the proposed approach using HMM in VR is presented. In Section 4, implementation details are provided in order to specify the parameters of the HMM. Section 5 presents the results of the full-body recognition. These results are discussed in Section 6. Finally, Section 7 concludes this paper.

2 Related Work

In the field of motion detection in VR there exist already a number of applications. However, to the best of our knowledge, none of this work uses the Vive Tracker in conjunction with supervised machine learning, such as HMM.

2.1 Motion Recognition in VR

Chan et al. [3] used a motion capture suit to follow the movements of the virtual teacher in order to learn how to dance. The suit with several attached markers

[2] http://optitrack.com, last visited on May 29th, 2018
[3] https://vicon.com, last visited on May 29th, 2018

can capture the full-body movements. The resulting data is then compared with "correct" reference data to give the student feedback about the correctness of her or his movement. Similar, Jiang et al. [11] presented a novel action recognition algorithm based on neural network. The researchers could reconstruct various full-body motions, such as walking, jogging, jumping, crouching and turning by only using Vive HMD and two associated Controllers.

Motion recognition in VR can also be done using a single Kinect sensor. The developers can easily exploit Microsoft Visual Gesture Builder NUI tool[4] in order to perform real-time gesture detection. Past work showed, that Kinect can be utilized to recognize motions in VR, e.g., for remote posture guidance during sports [10], for efficient strength training [22] or for efficiently dancing training [23]. In the approach proposed by Lee et al. [16], posing and acting is used as input to personalize furniture. The users can specify dimensions with simple speech commands while indicating a distance with their arms. However, speech recognition was performed using a trained operator, thus no machine learning approach was applied. Since the field of view of the Kinect is small, Rhodin et al. [21] attached a pair of fisheye cameras to the VR headset to increase the visibility of body parts in order to estimate the full-body pose. Furthermore, hand gestures can be recognized using a Leap Motion[5] device [6] or a Myo[6] armband [9, 15] to enable natural interaction in VR.

Furthermore, IMUs can be advantageous for motion detection. Fitzgerald et al. [7] integrated nine Xsens inertial-based sensors[7] to develop a biofeedback system for the instruction and analysis of sport rehabilitation exercises. However, the sensor data were stored in order to do offline analysis by comparing the data of the player and the expert. Another study used integrated sensors of a HMD to detect steps in real-time by applying a Gaussian low-pass filter [2]. With the recognition algorithm, the avatar's feet could be synchronized with the user's feet while walking on a treadmill.

2.2 Hidden Markov Models

Previous work showed that machine learning approaches and more specifically HMMs are widely and successfully applied in the field of speech and hand gesture recognition. First work has been already presented in the 80's and 90's, e.g., for automatic speech [19] or handwriting [24] recognition. Later on, Chen et al. [4] developed a real-time system to recognize hand gesture from a 2D video input. To reduce the complexity of hand detection in the images, markers were used in order to separate the hand from the background [14]. Similar, in the study conducted by Just and Marcel [12], the user had to wear two colored

[4] https://docs.microsoft.com/en-us/previous-versions/windows/kinect/dn785304(v%3dieb.10), last visited on May 25th, 2018
[5] https://www.leapmotion.com, last visited on May 28, 2018
[6] https://www.myo.com, last visited on May 28th, 2018
[7] https://www.xsens.com/functions/human-motion-measurement/, last visited on May 28th, 2018

gloves to facilitate hand tracking while recognizing two-handed gestures. Kao and Fahn [13] proposed an approach, where a single hand or both hands, as well as the face, have to be detected in each video frame in order to recognize hand gestures accurately. Hand gesture recognition can also be done based on recorded tracking data provided by a three-axis accelerometer embedded in a handheld device [18].

Other researchers collected body movement data with a Kinect in order to train the HMM [17, 20]. Even though the Kinect device is capable of tracking a full-body, both publications used this device to only recognize the upper body or more precisely to track gross arm movements.

3 Concept

The proposed recognition approach based on HMMs can be used to assist the player to learn a particular movement. A player can supervise own improvements. Another possible application example would be a training scenario where therapists can monitor the patient's recovery.

To determine the HMM, a teacher is needed to repeat the movement so that reference data can be gathered (ground truth). Since the recognition works regardless of the height of the teacher performing the posture [20], our model is generalized and independent of the size variance. Obtained data are then utilized to decide if the student performed the motion correctly. The VR environment provides visual and audible feedback to the student, informing her or him whether the executed movement was right or wrong.

3.1 Hidden Markov Model

The HMM was chosen since this approach showed good results for movement recognition in previous works (see Section 2). Motion recognition using HMMs essentially works in five steps (see Figure 1), as adapted from Yang and Xu [24].

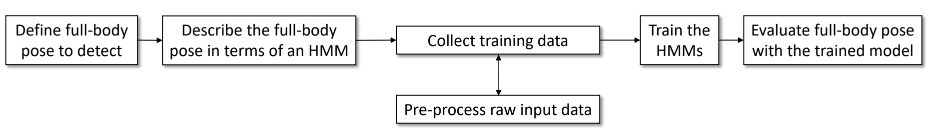

Fig. 1. Block diagram of the motion recognition-based system

In the *first step*, a pose or gesture, which should be recognized, must be defined. We want to show that our approach is able to recognize full-body motions in VR. Therefore, a pose must be selected, where the position of the hands and feet are important.

In the *second step*, the full-body pose must be described in terms of an HMM. The HMM is described as $\lambda = (S, V, A, B, \pi)$ as proposed by Rabiner [19]:

- $S = (s_1, ..., s_N)$, the set of N possible hidden states, i.e. possible values of q_i
- $V = (v_1, ..., v_M)$, the set of M possible observations, i.e. values of o_i
- $A \in \mathbb{R}^{N \times N}$, the transition matrix between the states, where the elements a_{ij} of the matrix indicate the probability for the transition from the state s_i to s_j, i.e. $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$
- $B \in \mathbb{R}^{N \times M}$, the distribution matrix, which indicates with which probability a state s_i leads to the observation v_j, i.e. $b_{ij} = b_i(j) = P(o_t = v_j | q_t = s_i)$
- $\pi \in \mathbb{R}^N$, the initial probability, i.e. a vector whose elements indicate with which probability s_i is the starting state: $\pi_i = P(q_1 = s_i)$

In this step only the structure of A and B are determined. The values of elements in A and B will be first estimated in the training process (Step 4). An overview is given in Figure 2.

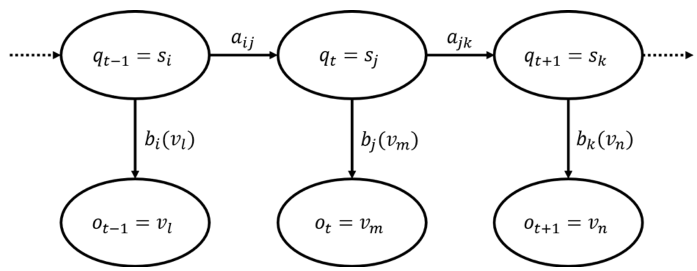

Fig. 2. Schematic representation of an HMM

In the *third step*, before the training data is collected, the raw input data has to be pre-processed. In order to train the HMM, several executions of the correct movement must be recorded and saved in a file. Since the HTC Vive provides global positions of the VR devices, this data has to be first transformed into a local coordinate system. Otherwise, the result would be distorted. Hence, the local orientation and position vector of all VR devices have to be continuously stored while the actual motion is being performed. Because the motion detection should work regardless of body height and execution speed, the data must be furthermore normalized.

In the *fourth step*, the HMMs should be trained using the collected data from the previous step. A console application, called *HMM-Trainer* allows the user to select training sequences and adjust the HMM parameters to train the model. The model parameters should be adjusted so that the likelihood $P(O|\lambda)$ can be maximized for the given training dataset. Therefore, the Baum-Welch algorithm is used to iteratively improve the values of λ to achieve the local maximum. In addition, the forward-backward algorithm to calculate the probability of a given HMM emitting a given observation sequence must be implemented.

Finally, in the *fifth step*, the full-body pose must be evaluated with the trained HMM. The trained model is then used to classify a full-body pose.

3.2 Feedback

In addition to the recognition of a full-body movement in VR, a constructive feedback is provided so that the student can improve the movement. Depending on whether the pose is recognized as correct or incorrect, the student gets audible feedback (e.g., "Your movement was correct" or "Your movement was wrong"). For a more detailed feedback about each individual Vive Controller/Tracker, a text is displayed on the PC and saved in a file. The student is informed whether the executed movement was right or wrong and if necessary, which position of the Controller/Tracker was recognized as correct or incorrect.

4 Implementation

The motion detection is implemented using the *Body Tracking Framework*[8] which was previously developed by KOM - Multimedia Communications Lab at the TU Darmstadt. To achieve maximal performance, the framework is based on the low-level game API and hardware abstraction library *Kore*[9] (`C++`). Audio files are created using IBM's speech synthesis software Watson[10] to provide audible feedback to the student. A demo can be viewed online[11].

```
N= 1242
HMD 41.8893 0.0018896 1.81965 0.00109858
lhC 41.8893 0.0708489 0.723469 0.122266
rhC 41.8893 -0.0718539 0.71499 -0.147675
bac 41.8893 0.207485 1.18198 0.160868
lfT 41.8893 0.227679 0.0898545 0.223854
rfT 41.8893 0.113464 0.112596 0.0269231
```

Fig. 3. An example of collected motion capture data. The first line indicates the number of data points. The followed rows indicate data points with a prefix, time and position vector (x, y, and z-axis).

First, the training data has to be collected by a teacher. When the virtual environment is loaded, the sensor data for the training of an HMM is recorded by pressing the "trigger" button on the Controller. The data is stored in a text file, as shown in Figure 3. Each line indicates positional data of a VR device at a certain time. The prefix at the beginning of the line is an identifier to associate the data with the VR devices: "lhC" and "rhC" correspond either the left or right Controller held in the hands, "bac" corresponds to the Tracker on the lower back (or hip), and "lfT" as well as "rfT" corresponds either to the left or right Tracker attached to the foot. In each row, time in seconds (equal for all six entries) and values for x, y, and z-axis are saved. The x and z-axis are defined

[8] https://github.com/CatCuddler/BodyTracking, last visited on May 22th, 2018
[9] https://github.com/Kode/Kore/, last visited on May 22th, 2018
[10] https://text-to-speech-demo.ng.bluemix.net, last visited on May 22th, 2018
[11] https://youtu.be/q-yKLtrTodA, last visited on May 30th, 2018

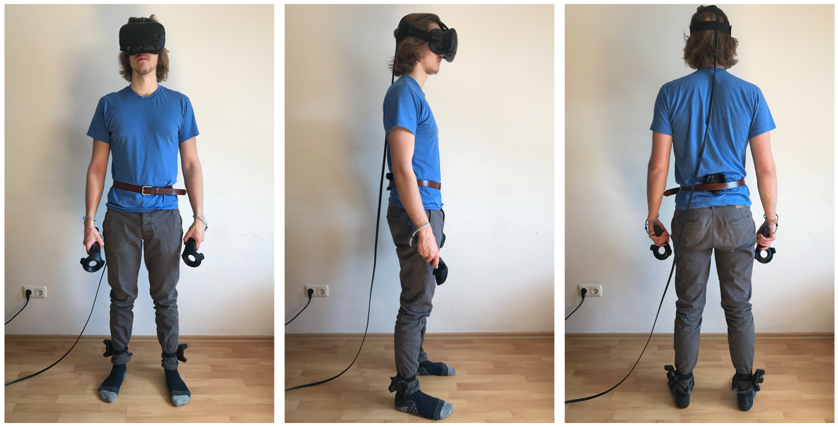

Fig. 4. Attached HTC Vive Tracker to the body.

as horizontal and the y-axis vertical. In addition, the number of data points is saved in the first line.

For training and evaluation, the Vive Trackers have to be attached to the teacher's and student's body. As it can be seen in Figure 4, two of the three additional Trackers are attached on the outside of the legs right above the ankle. It should be ensured that the Trackers are mounted as tight as possible in order to avoid falsification of the data by shaking. The third Tracker is fastened with a belt at the lower back. The Controllers are held in the hands and the HMD is worn on the head.

To define the HMM, we have to characterize the number of hidden states, the number of clusters, and the depth of the left-to-right matrix. We use *k-means clustering* to partition the observations of the data matrix into k cluster. For the initial estimation of the parameters, a teacher performed 17 correct and 30 incorrect versions of the yoga *Warrior I* pose. We deliberately chose this pose because it is well defined (only one correct possible execution). In other words, with another teacher, the correct movement should look very similar. The data was divided into a training and test set. We observed the result data of 6 to 16 hidden states, 8 to 100 cluster with random initial parameters and bounded left-to-right depth from 1 to 3. The HMMs were trained in order to calculate the log probability. By analyzing these log probabilities and standard deviations, some quantities could be calculated to adjust the HMM parameters. The first results of the test set indicate, that the smaller number of clusters (between 8 and 12) achieved better results. As expected, our model is overfitting due to a larger number of clusters approaching the number of actual data points (measurement values, i.e. position vector). The number of states, however, does not seem to have much influence. With six or eight hidden states the results are slightly better. Observing the left-to-right depth, the completely random initial parameters (depth of 0) yielded worse results. By increasing the depth parameter (i.e. depth of 2 or 3), better results were achieved.

5 Results

The HMM should be used to demonstrate that our concept is able to recognize complex full-body movements. To complete the pose correctly (see Figure 5), the student starts the movement while standing and takes a step backward with his left foot. Both heels should lie on one line and the rear foot should be turned outwards. Then the arms are stretched out in front of the body and the upper body remains upright. The right knee should now be directly above the right heel. Finally, the arms are stretched out parallel to each other with the palms facing inwards.

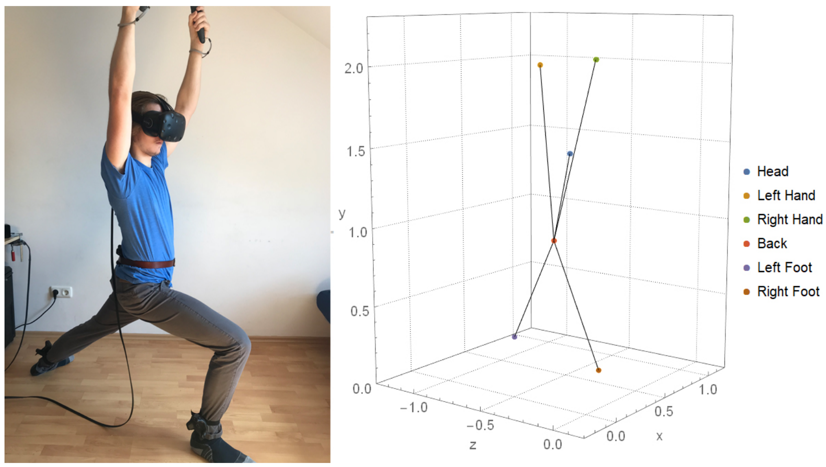

Fig. 5. The final position of the yoga *Warrior I* pose (left) and visualization of the test data (right).

5.1 Training

In the initial tests (see Section 4), no explicit ideal HMM parameters could be determined. Therefore, a total number of further 100 correct movements of the yoga *Warrior I* pose were collected (training set). Two final HMMs were trained and finally tested to find the best model. The first HMM has six hidden states (HMM6) and the second HMM 10 hidden states (HMM10). For HMM6 we use 8 clusters and left-to-right depth of 2 while for the HMM10 100 clusters and left-to-right depth of 3 are used.

5.2 Testing

To test the final HMMs, a test set of 40 correct movements and 40 incorrect movements was used. We furthermore divided the test set of the incorrect movements into 20 partially correct or similar movements (e.g., only the legs were correctly positioned, however, the arms were stretched to the side rather than up or the arms were correctly positioned and the legs not) and 20 totally incorrect movements (e.g., either legs nor arms were correctly positioned). Here the motions are recognized in real-time and the student gets audible feedback right after the motion is completed. The results can be obtained from the Table 1. For the evaluation, we furthermore introduced the second condition. We wanted to evaluate if false positives are recognized when the Tracker/Controller do not move during the exercise. Thus, HMM6* and HMM10* will recognize a movement as wrong as soon as one Tracker/Controller remains in the same cluster during the entire movement.

Table 1. The results with the four final HMMs. In each case, the probability of correctly recognized movements (either right or wrong) is given.

	Positive	Similar	Negative	**Total**
HMM6	97,5% (39 / 40)	50% (10 / 20)	100% (20 / 20)	86.25%
HMM6*	90% (36 / 40)	75% (15 / 20)	100% (20 / 20)	88.75%
HMM10	0% (0 / 40)	100% (20 / 20)	100% (20 / 20)	50%
HMM10*	0% (0 / 40)	100% (20 / 20)	100% (20 / 20)	50%

The results clearly show that neither HMM10 nor HMM10* is suitable for detecting *Warrior I* movement, as none of the correct motions are detected. The reason for this is that the current implementation of the forward algorithm is not robust to underflow. Thus, even with the correct (*positive*) movements and despite the fact that input data is normalized, the HMM10 is not able to recognize the movements. As described by Blunsom [1] we could avoid underflow by using a scaling coefficient c_t, that only depends on t and can keep the probability values in the dynamic range of the machine.

A significantly better result is provided by HMM6 or even HMM6*. These two models can recognize all completely wrong (*negative*) movements. Analyzing the partially correct movements (*similar*), the recognition rate of HMM6 drops to 50%. With HMM6*, the accuracy of similar poses is increased to 75%. Compared to the HMM6 (with the accuracy of 97,5%), the detection of the correct movement decreases to 90% when using HMM6*. We believe this happens because the right foot and back do not move much while performing *Warrior I* pose. Therefore, HMM6 should be used for movements, where the positions of all extremities differ much between the initial and final position. For other movements, HMM6* should be used.

6 Discussion

The results presented in the previous Section show that the proposed concept can be used to recognize a complex pose. Therefore, we believe that our method can also be used to detect additional full-body movements: it can be used for exergames where the player is motivated to move or to learn a certain movement pattern. The results suggest that our concept is able to detect complex movements, such as yoga *Warrior I* pose. This movement takes advantage of all existing sensors (Vive HMD, two Controllers, and three Tracker). We demonstrated that a full-body movement can be detected with a probability of over 88%. We believe that our system is able to recognize various full-body movements in VR.

Compared to related work using HMM, we could obtain similar results. In the evaluation conducted by Chen et al. [4] an accuracy of 85% was achieved, while Raheja et al. [20] achieved a recognition success rate of over 90%. Other researchers detected more than 95% of the movements [13, 14, 17, 18] or even more than 99% [24] as correct. The step detection algorithm using a Gaussian low-pass filter provided an accuracy of up to 98.7% while walking at different speeds [2]. Recognition of various hand gestures using Support Vector Machine reached an accuracy of 80%[12]. However, in most publications, only motions of one body part (e.g., only hand gestures or steps) were recognized and not a full-body pose as in our approach. Furthermore, some works also only compare gestures and do not take the detection of wrong movements into account.

To make the current HMM reliable in terms of player's execution speed and height, we normalize the tracking data. To improve our approach, a *Dynamic Time Warping* could be furthermore used, as the one proposed by Raheja et al. [20]. Alternatively, a promising approach of *Conversive Hidden non-Markovian Models* has recently been developed that can explicitly consider information about the speed of a movement [5]. With this approach, it would be possible to recognize movements which differ in execution speed even more accurately. For further improvements, calibration may also be necessary.

Additionally, since we want to reduce the sensor amount (currently using five VR tracking devices in addition to the HMD), only the position and orientation of the extremities (e.g., hands and feet) are known. To additionally track other joints (e.g., elbow and knee) we do not need to increase the number of sensors. By solving the Inverse Kinematics (IK), the position and orientation of every joint can be estimated. Using the IK solution we could distinguish between different full-body movements, where the positions/orientations of a large number of joints are crucial.

7 Conclusion

In this paper, we present a concept to recognize full-body movements in VR using only a small number of VR devices: a HTC Vive HMD, two Vive Controllers and three Vive Trackers which have to be attached to the body. Thus, the head, both hands, and both feet, as well as the upper body, are tracked. The positions of all these body parts are important for the motion recognition. Furthermore, the students get immediate feedback about the motion performance and can thus

[12] https://tzuchanchuang.itch.io/gesture-recognition-input-method-for-ar, last visited on June 6th, 2018

easily improve their execution. Further work could include extended feedback, describing the detailed improvements for the movement.

Our approach allows to train the model for various full-body gestures/poses easily and without much extra effort. The trained model is independent of the player's movement speed and it provides similar results regardless of the player's height. The evaluation showed that we can recognize a full-body movement, such as yoga *Warrior I*, with an accuracy of 88.75%. However, we evaluated our HMM with only one pose. In the future work, various additional movements should be trained, e.g. dancing or gestures for locomotion (walking in place, swimming).

Further research will focus on optimizing HMM parameters to improve the average recognition rate. We also want to improve the VR experience. For a better visual performance, a virtual mirror could be used so that the student can see her or his movements as well as the movements of the teacher. The movements can be represented through an avatar using IK approaches to provide real-time feedback.

References

1. Blunsom, P.: Hidden Markov Models. Lecture Notes, August **15**, 18–19 (2004)
2. Caserman, P., Krabbe, P., Wojtusch, J., von Stryk, O.: Real-Time Step Detection Using the Integrated Sensors of a Head-Mounted Display. In: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC). pp. 3510–3515 (Oct 2016)
3. Chan, J.C.P., Leung, H., Tang, J.K.T., Komura, T.: A Virtual Reality Dance Training System Using Motion Capture Technology. IEEE Transactions on Learning Technologies **4**(2), 187–195 (April 2011)
4. Chen, F.S., Fu, C.M., Huang, C.L.: Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models. Image and Vision Computing **21**(8), 745 – 758 (2003)
5. Dittmar, T., Krull, C., Horton, G.: A New Approach for Touch Gesture Recognition: Conversive Hidden non-Markovian Models. Journal of Computational Science **10**, 66 – 76 (2015)
6. Ferracani, A., Pezzatini, D., Bianchini, J., Biscini, G., Del Bimbo, A.: Locomotion by Natural Gestures for Immersive Virtual Environments. In: Proceedings of the 1st International Workshop on Multimedia Alternate Realities. pp. 21–24. AltMM '16, ACM (2016)
7. Fitzgerald, D., Foody, J., Kelly, D., Ward, T., Markham, C., McDonald, J., Caulfield, B.: Development of a Wearable Motion Capture Suit and Virtual Reality Biofeedback System for the Instruction and Analysis of Sports Rehabilitation Exercises. In: 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. pp. 4870–4874 (Aug 2007)
8. Han, J., Shao, L., Xu, D., Shotton, J.: Enhanced Computer Vision with Microsoft Kinect Sensor: A Review. IEEE Transactions on Cybernetics **43**(5), 1318–1334 (Oct 2013)
9. Han, P.H., Huang, D.Y., Tsai, H.R., Chen, P.C., Hsieh, C.H., Lu, K.Y., Yang, D.N., Hung, Y.P.: Moving Around in Virtual Space with Spider Silk. In: ACM SIGGRAPH 2015 Emerging Technologies. pp. 19:1–19:1. SIGGRAPH '15, ACM, New York, NY, USA (2015)

10. Hoang, T.N., Reinoso, M., Vetere, F., Tanin, E.: Onebody: Remote Posture Guidance System Using First Person View in Virtual Environment. In: Proceedings of the 9th Nordic Conference on Human-Computer Interaction. pp. 25:1–25:10. NordiCHI '16, ACM, New York, NY, USA (2016)
11. Jiang, F., Yang, X., Feng, L.: Real-time Full-body Motion Reconstruction and Recognition for Off-the-shelf VR Devices. In: Proceedings of the 15th ACM SIGGRAPH Conference on Virtual-Reality Continuum and Its Applications in Industry - Volume 1. pp. 309–318. VRCAI '16, ACM, New York, NY, USA (2016)
12. Just, A., Marcel, S.: A Comparative Study of two State-of-the-Art Sequence Processing Techniques for Hand Gesture Recognition. Computer Vision and Image Understanding **113**(4), 532 – 543 (2009)
13. Kao, C.Y., Fahn, C.S.: A Human-Machine Interaction Technique: Hand Gesture Recognition Based on Hidden Markov Models with Trajectory of Hand Motion. Procedia Engineering **15**, 3739 – 3743 (2011), cEIS 2011
14. Keskin, C., Erkan, A., Akarun, L.: Real Time Hand Tracking and 3D Gesture Recognition for Interactive Interfaces Using HMM. ICANN/ICONIPP **2003**, 26–29 (2003)
15. Koniaris, B., Huerta, I., Kosek, M., Darragh, K., Malleson, C., Jamrozy, J., Swafford, N., Guitian, J., Moon, B., Israr, A., Mitchell, K.: IRIDiuM: Immersive Rendered Interactive Deep Media. In: ACM SIGGRAPH 2016 VR Village. pp. 11:1–11:2. SIGGRAPH '16, ACM, New York, NY, USA (2016)
16. Lee, B., Cho, M., Min, J., Saakes, D.: Posing and Acting as Input for Personalizing Furniture. In: Proceedings of the 9th Nordic Conference on Human-Computer Interaction. pp. 44:1–44:10. NordiCHI '16, ACM, New York, NY, USA (2016)
17. Nguyen-Duc-Thanh, N., Lee, S., Kim, D.: Two-Stage Hidden Markov Model in Gesture Recognition for Human Robot Interaction. International Journal of Advanced Robotic Systems **9**(2), 39 (2012)
18. Pylvänäinen, T.: Accelerometer Based Gesture Recognition Using Continuous HMMs. In: Iberian Conference on Pattern Recognition and Image Analysis. pp. 639–646. Springer (2005)
19. Rabiner, L.R.: A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. Proceedings of the IEEE **77**(2), 257–286 (Feb 1989)
20. Raheja, J., Minhas, M., Prashanth, D., Shah, T., Chaudhary, A.: Robust Gesture Recognition Using Kinect: A Comparison Between DTW and HMM. Optik - International Journal for Light and Electron Optics **126**(11), 1098 – 1104 (2015)
21. Rhodin, H., Richardt, C., Casas, D., Insafutdinov, E., Shafiei, M., Seidel, H.P., Schiele, B., Theobalt, C.: EgoCap: Egocentric Marker-less Motion Capture with Two Fisheye Cameras. ACM Trans. Graph. **35**(6), 162:1–162:11 (Nov 2016)
22. Tanaka, Y., Hirakawa, M.: Efficient Strength Training in a Virtual World. In: 2016 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-TW). pp. 1–2 (May 2016)
23. Yan, S., Ding, G., Guan, Z., Sun, N., Li, H., Zhang, L.: OutsideMe: Augmenting Dancer's External Self-Image by Using A Mixed Reality System. In: Proceedings of the 33rd Annual ACM Conference Extended Abstracts on Human Factors in Computing Systems. pp. 965–970. CHI EA '15, ACM, New York, NY, USA (2015)
24. Yang, J., Xu, Y.: Hidden Markov Model for Gesture Recognition. Tech. rep., Robotics Institute, Carnegie-Mellon University, Pittsburgh (May 1994)