MODELING OF GESTURES WITH DIFFERING EXECUTION SPEEDS: ARE HIDDEN NON-MARKOVIAN MODELS APPLICABLE FOR GESTURE RECOGNITION

Sascha Bosse[(a)], Claudia Krull[(b)], Graham Horton[(c)]

[(a)(b)(c)] Otto-von-Guericke-University Magdeburg
P.O. Box 4120
39016 Magdeburg, Germany

[(a)]sbosse@st.ovgu.de, [(b)]claudia.krull@ovgu.de, [(c)]graham.horton@ovgu.de

ABSTRACT
Gesture recognition is an important subtask of systems implementing human-machine-interaction. Hidden Markov Models achieve good results for gesture recognition in real-time supporting a low error rate. However, the distinction of gestures with different execution speeds is difficult. Hidden non-Markovian Models provide an approach to model time dependent state transitions to eliminate these problems. In this paper, a basic non-Markovian model structure for gesture recognition is developed. The experiments show that Hidden non-Markovian Models are not only applicable in the field of gesture recognition, but that they can also distinguish gestures with different execution speeds.

Keywords: Gesture recognition, Hidden non-Markovian Models

1. MOTIVATION
Automatic gesture recognition plays a key role in human-machine-interaction within virtual environments and multi-modal feedback systems. To comply with the requirements of such systems, the recognition must be performed in real-time and with a small number of errors (Rigol, Kosmola, and Eickeler 1997). Hidden Markov Models (HMMs), successfully applied in the field of pattern recognition, provide satisfactory results that meet these conditions (Frolov, Deml, and Hanning 2008; Rigol, Kosmola, and Eickeler 1997; Chen, Fu, and Huang 2003).

However, by definition, time dependent processes cannot be described easily in these models. A differentiation of gestures with varying speeds of execution is hardly realizable. This could be useful to reduce the number of relevant gestures or to classify gestures by their execution speed to optimize the performance of movements, for example in a virtual environment.

Hidden non-Markovian Models (HnMMs) were developed to support the easy modeling and fast execution of hidden models without the restriction of memoryless state transitions. For this purpose, to each state transition an arbitrary distribution function can be assigned. The paper will therefore investigate whether HnMM can distinguish gestures with different execution speeds.

2. GOAL AND TASKS
The first goal of this paper is to determine whether HnMMs are applicable in the area of gesture recognition in real-time with a similar error rate as provided by HMMs. For this purpose a non-Markovian model will be tested on a standard-PC and should recognize 90% of the executed gestures.

If this goal is achieved, the question remains whether the non-Markovian approach is able to distinguish gestures with differing execution speeds. The second goal of this paper is to determine that and to compare the results with those of the Markovian approach.

The following tasks were identified to reach the goals:

1. Development of an HnMM structure that models gestures
2. Selection of a gesture catalog to test these models
3. Comparison of the HnMMs' result with those of HMMs

3. APPROACH
Existing approaches to gesture recognition using HMMs may be divided into two categories: Recognition from image data and recognition from data gloves. The latter approach, in a simplified form, will be pursued here. For this purpose a so-called Wiimote (Figure 1), a remote control of the 2006 published Nintendo Wii, will be used. It can gather motion data in a triple-axis space.

Figure 1: Wii Remote

A great advantage of HMMs is the ability to automatically train models to improve the performance successively. Unfortunately this cannot be done yet with HnMMs, so the developed models must be trained manually (Krull and Horton 2009). Due to the fact that this procedure is very time-consuming, only a small number of gestures will be selected.

4. RELATED WORK

Many papers deal with the application of HMMs in the field of gesture recognition. In the following, some examples of those papers and their results will be highlighted.

Frolov, Deml, and Hanning (2008) investigate the ability of HMMs to improve multi-modal haptic feedback. It is shown that this approach can predict what physical attributes a user wants to know, recognized from his gestures. For that purpose, eight gestures were chosen, which differ in their muscular movement (recorded by a P5-Glove), but not in their execution speed.

In (Rigol, Kosmola, and Eickeler 1997), a real-time system for gesture recognition from image data is presented. It uses HMMs for 24 gestures with a recognition rate of 92.9%. Every state in the model represents a picture frame of the input data. This provides the ability to execute a gesture at any speed, but not the distinction of different speeds.

Chen, Fu, and Huang (2003) present an approach where image data is processed to a spatial and temporal feature vector using the Fourier descriptor and motion analysis. The temporal features are used to enable time-varying hand shapes while recognizing a gesture. The variances in speed are not considered.

Lee and Kim (1999) focus on the recognition of non-gesture hand motion to determine when an important gesture starts and when it stops. An artificial threshold model is used, so that any possible gesture can be described by it. Thus, the likelihood of the dedicated model for this gesture will be highest. The tested gestures are characterized by different spatial motions while the differences in execution speeds are not integrated into the results.

The selected works show that Hidden Markov Models are successfully used to recognize gestures in various different contexts. But the ability to distinguish gestures at different execution speeds was never considered. The question remains whether Markovian models are not applicable in this context or whether it simply has not been tested yet.

In the following, two approaches (one Markovian and one non-Markovian) are modeled to examine the ability of gesture recognition and distinction.

5. MODELING

The first step to achieve the goals is to bring the data stream of the Wii Remote in a format that can be processed further. Data from the Wii Remote is received nearly 100 times per second, so the values of the three dimensions of the acceleration sensor can be retrieved. Each value stands for the amount of gravity the Wiimote is exposed to in the three directions, normalized to a range from -1 to 1. Because of earth’s own gravity also the current orientation of the remote can be accessed. To remove noise from the signal, some kind of windowing is necessary. For this purpose, every ten measurements a mean value is computed, which will be processed for the different models.

5.1. Signal Outputs

In the HMM case, the so formed continuous data stream will be interpreted as the signal output trace of the HMM. Therefore, the values are categorized in specific ranges representing the discrete signal values. These ranges are: *large negative acceleration* [-1,-0.5], *small negative acceleration* [-0.5,-0.1], *no acceleration* [-0.1,0.1], *small positive acceleration* [0.1,0.5] and *large positive acceleration* [0.5,1].

A similar procedure is applied to create traces for the non-Markovian case, but, only significant changes in this data stream are used as trace signals of the model. The next step is to develop basic model classes for the two approaches.

5.2. Markovian Model Structure

To perform a gesture, three phases are passed. The first is to bring the hand in the initial position of the gesture, while the second phase is to execute the different atomic motions. In the last phase, the hand moves from the end to a neutral position. According to this process, the HMMs consist of four states: *Neutral*, *Start*, *Execution* and *Stop*. Figure 2 shows a simplified HMM (hiding the reflexive DTMC arcs) with possible output signals in each state. Each circle represents a state while the solid arrows represent the possible state changes. The arrows with the dashed line represent the possible output symbols of each state.

In the neutral state, only a specified orientation of the remote should be emitted. Every notable movement of the Wiimote should cause the model to change to the start state. If the next motions could be matched as parts of the gesture, the model will change to the execution state. In the other case, the model goes back to the neutral state or it remains in the start state. This procedure should guarantee that random motions do not influence the recognition of a gesture.

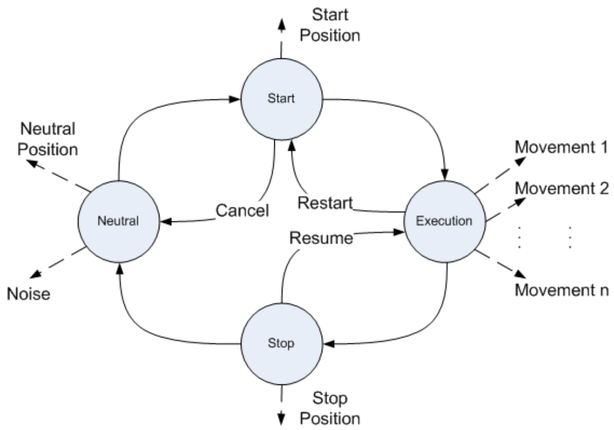

Figure 2: Simple HMM Model for Gesture Recognition

The execution of a gesture consists of different atomic movements, which should be emitted in the execution state. If the execution fails, the user could restart the gesture, so the model returns to the start state. When the remote's movement stops, the model changes to the stop state. In this state the execution can be resumed and further movements can be executed. After performing the gesture the Wiimote should return to its neutral position so that the model changes to the neutral state.

5.3. Non-Markovian Model Structure

For the non-Markovian models, a similar approach is developed. The basic idea of the gesture phases can be transferred, but because the output signals are emitted by transitions instead of states, only changes of the movement generate signal outputs. These changes are characterized by the variation of the acceleration values. This variation $v=acc(t)-acc(t-1)$ has a value range from -2 to 2, so the following classification is done: *large negative variation* [-2,-1], *medium negative variation* [-1,-0.5], *small negative variation* [-0.5,-0.1], *no variation* [-0.1,0.1], etc. Alternatively, a mixture of variation and current acceleration could be used.

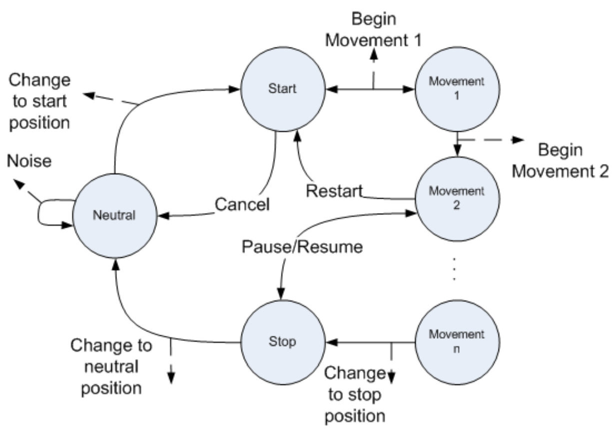

Figure 3: Abstract HnMM Model for Gesture Recognition

In contrast to the HMM, there are no discrete time steps in this model. Time progresses continuously and to each state transition a stochastic time distribution is assigned. In Figure 3 an abstract state space of the model is presented. Every state transition generates an output. Each circle represents a state and the solid arrows represent the possible state changes. The arrows with the dashed line represent the possible output symbols of each state change.

The model's state space is similar to the HMM, only the execution state is split into *n* movements. This is done for reasons of clarity. In the final model there will be only one execution state with *n* reflexive transitions. Only these transitions are not uniformly distributed. This should guarantee that the model likelihood depends only on the execution time for movements, not on time for irrelevant movement changes.

6. IMPLEMENTATION

After developing the conceptual models, the necessary functions can be implemented. These are:

1. Receive data from the Wii Remote,
2. Compute signal outputs,
3. Train the models and
4. Compute likelihood of Markovian and non-Markovian models.

To communicate with the Wiimote, the open source *Managed Library for Nintendo's Wiimote* (WiimoteLib) for .NET in version 1.7 is used (http://wiimotelib.codeplex.com/).

It provides an event which is fired every time the status of the Wiimote changes. In this event the data from the acceleration sensors can be processed. After smoothing the input (see last section) the different signal outputs for both model classes are generated. Depending on the program mode, they are discarded (normal mode), exported to a csv-file (train mode) or saved in lists (recognize mode).

6.1. Training and Evaluation of the HMM

To train the HMMs, firstly a movement is chosen. After this, the train mode must be activated and the movement must be carried out. The thus obtained data is used to train the Markovian models automatically, using the Baum-Welch Algorithm as described in (Fink 2008). The following open source implementation was used: *Hidden Markov Models in C#* project by César de Souza (http://www.codeproject.com/Articles/69647/Hidden-Markov-Models-in-Csharp.aspx). In addition to the training function this code also provides the evaluation of signal outputs with respect to the models (Fink 2008). So for every model the likelihood of the gesture having been carried out can be computed.

Because there are three dimensions of the acceleration data, five possible output values per dimension in the Markovian and seven output values in the non-Markovian case, the three dimensions are treated independently. That means that for every gesture there are three models trained only with the respective dimension data. If not, there would be 125 output symbols for the HMM (343 for the HnMM), which would slow down the computation and complicate the training of the non-Markovian models unnecessarily. However, to prevent that the assumption of full independence distorts the results too much, the three models are connected in a way that a state transition must be executed simultaneously in every model. The probability of emitting a special symbol tuple is computed by multiplying the three single output probabilities.

6.2. Training and Evaluation in the HnMM

To obtain the model parameters in the non-Markovian case, a manual training is carried out. For that purpose the created csv-files are checked. Every time a significant change of the acceleration sensors was recognized, the time stamp and the symbol outputs were recorded. With respect to the gesture carried out, the characteristic movement changes are identified as state transitions. The parameters of the transitions non-Markovian distributions are estimated from the time span between these movement changes. The symbol output probabilities are derived from the recorded symbols by their frequency in the particular csv-files.

After training the HnMMs the likelihood of a gesture must be computed. To reach the goal of real-time computation, an adaption of the established HMM algorithms is required. Since the here developed models fulfill some properties (every transition omits a symbol, only one transition can be fired between two states and no race age transitions are allowed), the original formula of the Forward-Algorithm (Fink 2008) to evaluate a given symbol sequence (see Equation (1)) can be adapted to Equation (2), where the probability of the state transition a_{ij} is replaced by the integral of the state change rate over the time elapsed since the last state change (Krull and Horton 2009). This rate corresponds in this case to the instantaneous rate function (IRF) shown in Equation (3), where $f(t)$ is the probability density function and $F(t)$ is the cumulative distribution function of the state change distribution from i to j over time (Horton 2002).

$$\alpha_{k+1}(j) = \sum_{i=1}^{N} \alpha_k(i) * a_{ij} * b_j(o_{k+1}) \tag{1}$$

$$\alpha_{t_{k+1}} = \sum_{i=1}^{N} \alpha_{t_k}(i) * \int_0^{t_{k+1}-t_k} a_{ij}(x)dx * b_{ij}(o_{k+1}) \tag{2}$$

$$a_{ij}(t) = \mu_{ij}(t) = \frac{f(t)}{1-F(t)} \tag{3}$$

To compute an approximation of the integral, the trapezoidal rule (see Equation (4)) is applied.

$$\int_a^b f(x)dx \approx (b-a)\frac{f(a)+f(b)}{2} \tag{4}$$

With the initialization of $\alpha_0 = \pi_i$, a recursion can be implemented which finally provides the sum of the probability of all possible paths, which can be interpreted as the likelihood of the model to generate the given symbol sequence.

When all likelihood values are computed, the results are compared and the best fitting model corresponds to the most likely gesture. So now the evaluation results of the Markovian and the non-Markovian models can be compared for the chosen gestures.

7. EXPERIMENTS AND RESULTS

7.1. Experiment Description and Expectations

To compare the two different approaches, some experiments will be carried out. To this end, gestures are chosen and the corresponding models are created. While performing these gestures, the output will be saved, so the models can be trained. Afterwards both models should recognize the gesture when it is executed in real-time. This whole procedure is done with one gesture in two execution speeds. If the non-Markovian model can distinguish these two speeds while the Markovian cannot, the first model is a better approach for this purpose.

This comparison is planned for one simple and one more complex gesture. The simple one is a movement upwards. Executing this at different speeds could be used to distinguish a scrolling move from a “to-the-top-move”. The more complex gesture is a sequence of the atomic movements left, right, up.

For the four gestures up fast (*Uf*), up slow (*Us*), left-right-up fast (*LRUf*) and left-right-up slow (*LRUs*), The Hidden Markov Models are trained with ten csv-files until the difference between the likelihood of two consecutive iterations is smaller than $1*10^{-5}$. As described in the above section, the non-Markovian models are trained manually.

7.2. Initial Experiment

To test the trained models, every gesture was executed ten times and the most likely gesture was recorded. Table 1 shows the results of the Markovian and Table 2 the results of the non-Markovian models. The bold numbers mark a correct recognition. In this initial experiment, the HMMs provide a recognition rate of only 50%, the HnMMs a rate of 75% over all performed gestures.

Table 1: HMM Results of Initial Experiment

Reco.	Exec.	Uf	Us	LRUf	LRUs
		0	0	20	20
Uf	10	**0**	0	10	0
Us	10	0	**0**	0	10
LRUf	10	0	0	**10**	0
LRUs	10	0	0	0	**10**

Table 2: HnMM Results of Initial Experiment

Reco.	Exec.	Uf 10	Us 17	LRUf 9	LRUs 4
Uf	10	**8**	2	0	0
Us	10	0	**10**	0	0
LRUf	10	2	0	**8**	0
LRUs	10	0	6	0	**4**

While the results of the non-Markovian models are acceptable, the classical models failed abnormally. Table 1 indicates that the atomic gesture *Up* cannot be distinguished from the more complex gesture *Left-Right-Up*.

After closer examination, it was found that a property of the Wiimote's acceleration sensor is responsible: Because of earth's own gravity, the sensor's z-axis is much more sensible to changes than the other dimensions. Since the left-right-movements left the Wiimote in its initial alignment with respect to the ground these movements were identified as a kind of noise, thus the distinction of the two gestures failed.

7.3. Experiment with Adapted Gestures

To make the different gestures more easily distinguishable, the two gestures LRUf and LRUs are changed, so that the left and right movements are supported by a rotation. Therefore, the effect of the gravity should be recognized in other dimensions as well.

The result of the experiment with the adapted gestures are shown in Table 3 for the HMM and Table 4 for the HnMM. They show a recognition rate of 85% in the Markovian case and 88% in the non-Markovian case.

Table 3: HMM Results with Adjusted Gestures

Reco.	Exec.	Uf 5	Us 9	LRUf 15	LRUs 11
Uf	10	**5**	0	5	0
Us	10	0	**9**	0	1
LRUf	10	0	0	**10**	0
LRUs	10	0	0	0	**10**

Table 4: HnMM Results with Adjusted Gestures

Reco.	Exec.	Uf 8	Us 14	LRUf 11	LRUs 7
Uf	10	**8**	2	0	0
Us	10	0	**10**	0	0
LRUf	10	0	0	**10**	0
LRUs	10	0	2	1	**7**

The results are now in line with the expectations. The non-Markovian model shows slightly better results, because the Markovian one still cannot completely distinguish the *Up* from the *Left-Right-Up* gesture. The errors in the non-Markovian case can be explained by the lack of robustness induced by the manual training.

However, contrary to our expectations, the distinction of slowly and fast executed gestures can be done by both models. A possible cause for this behavior could be that the chosen symbol sequences, the acceleration sensor values, are indicators of the execution speed, because their actual values strongly depend on the intensity of a movement. To eliminate this behavior, the symbol sequences are adapted so that they only specify whether a value is positive, neutral or negative.

7.4. Experiment with Reduced Output Symbol Set

To reduce the number of possible output signals (and also their information value), the discretisation intervals are adjusted as follows: In the HMM case, the signals are *negative acceleration* [-1,-0.1], *no acceleration* [-0.1,0.1] and *positive acceleration* in [0.1,1]. In the HnMM case, they are *negative variation* [-2,-0.1], *no variation* [-0.1,0.1] and *positive variation* [0.1,2].

After this reduction a new training is performed. The results of the following experiments are presented in Table 5 and in Table 6 (non-Markovian). In this sample the HMMs recognize 55% and the HnMMs 80% of the executed gestures. After a closer look on the results it becomes obvious that the non-Markovian model's behavior is similar to the last test series. The Markovian model however is no longer capable of distinguishing fast and slow gestures correctly.

Table 5: Final HMM Results

Reco.	Exec.	Uf 6	Us 3	LRUf 19	LRUs 12
Uf	10	**6**	0	3	1
Us	10	0	**3**	5	2
LRUf	10	0	0	**7**	3
LRUs	10	0	0	4	**6**

Table 6: Final HnMM Results

Reco.	Exec.	Uf 7	Us 12	LRUf 16	LRUs 5
Uf	10	**7**	0	3	0
Us	10	0	**10**	0	0
LRUf	10	0	0	**10**	0
LRUs	10	0	2	3	**5**

To illustrate this conclusion, the normalized likelihoods of two exemplary gestures are shown. Figure 4 shows the likelihoods of all eight models when a *Left-Right-Up Fast* movement is executed, which was recognized correctly by both models. Figure 5 shows the likelihoods of all eight models when an *Up Slow* movement is executed, which was only recognized by the HnMM. These diagrams show that the recognition is clear in the non-Markovian case but ambiguous in the Markovian case.

So in the end the Hidden non-Markovian Models provide better results in the differentiation of gestures' execution speed if the decoded signal outputs are not an indicator of the speed of the gesture.

Figure 4: Normalized Likelihood of LRUf

Figure 5: Normalized Likelihood of Us

8. CONCLUSION

In this paper it was shown that Hidden non-Markovian Models are applicable in the field of gesture recognition. We could also demonstrate that, under certain circumstances, non-Markovian models provide a better distinction of gestures with different execution speeds than Markovian models do. So the goals of this paper are reached. Further it was shown that it can be necessary to discard a Markovian model because the ability of gesture distinction is not sufficient. Hidden non-Markovian Models represent an alternative approach to solve this problem.

So it is feasible to introduce time dependent state transitions to distinguish similar signal sequences by their output speed. This opens up opportunities to model transitions with arbitrary stochastic distributions in applications with hidden states. Adding meaning to the execution speed of a gesture can reduce the gesture catalogue, which a user has to learn. Looking ahead it can be postulated, that HnMMs are able not only to decide whether an arbitrary pattern was executed correctly, but also whether it was executed at the right speed.

The main disadvantage of HnMMs is the manual training process. This includes the development of the basic model and the computation of state change distributions and output probabilities. In addition to the large time effort, this method also produces less robust models than automatic learning algorithms do. This is still an area of future work and must be automated somewhat to make the approach feasible in real applications.

Apart from the modeling approach it is also shown that the usage of the Wii Remote in the field of gesture recognition has disadvantages. The built-in acceleration sensor is very sensitive to rotation movements but not that sensitive to translation movements. Maybe a combination of acceleration sensor and position tracking with the infrared camera could eliminate this problem.

REFERENCES

Chen F.; Fu C.; and Huang C., 2003. Hand gesture recognition using a real-time tracking method and hidden Markov models. *Image and Vision* Computing 21. 745-758.

Fink G.A., 2008. Markov Models for Pattern Recognition. Springer Verlag. ISB 978-3-540-71766-9.

Frolov V.; Deml B.; and Hanning G., 2008. Gesture Recognition with Hidden Markov Models to Enable Multi-modal Haptic Feedback. In EuroHaptics 2008. 786-795.

Horton G., 2002. A new paradigm for the numerical simulation of stochastic petri nets with general firing times. In Proceedings of the European Simulation Symposium 2002. 129-136.

Krull C. and Horton G., 2009. HIDDEN NON-MARKOVIAN MODELS: FORMALIZATION AND SOLUTION APPROACHES. In Proceedings of 6th Vienna International Conference on Mathematical Modelling. 682-693.

Lee H. and Kim J., 1999. An HMM-Based Threshold Model Approach for Gesture Recognition. In IEEE Transactions on Pattern Analysis and Machine Intel- ligence 21. 961-973.

Rigol G.; Kosmola A.; and Eickeler S., 1997. High Performance Real-Time Gesture Recognition Using Hidden Markov Models. In Lecture Notes in Computer Science Vol. 1371. 69-80.