Real Time Gesture Learning and Recognition: Towards Automatic Categorization

Jean-Baptiste Thiebaut[†], Samer Abdallah[‡], Andrew Robertson[‡],
Nick Bryan Kinns[†], Mark Plumbley[‡]
†Interaction, Media and Communication Group, {jbt,nickbk}@dcs.qmul.ac.uk
‡Centre for Digital Music, {samer.abdallah,andrew.robertson,mark.plumbley}@elec.qmul.ac.uk
Queen Mary, University of London

ABSTRACT

This research focuses on real-time gesture learning and recognition. Events arrive in a continuous stream without explicitly given boundaries. To obtain temporal accuracy, we need to consider the lag between the detection of an event and any effects we wish to trigger with it. Two methods for real time gesture recognition using a Nintendo Wii controller are presented. The first detects gestures similar to a given template using either a Euclidean distance or a cosine similarity measure. The second method uses novel information theoretic methods to detect and categorize gestures in an unsupervised way. The role of supervision, detection lag and the importance of haptic feedback are discussed.

Keywords

Gesture recognition, supervised and unsupervised learning, interaction, haptic feedback, information dynamics, HMMs

1. INTRODUCTION

Gesture forms an integral part of music performance. Traditional instrumentalists develop a virtuosity for the gestures related to their instruments. In a similar manner, the performers who use digital interfaces develop a virtuosity adapted to their devices, and an important issue to address is to categorize and recognize these gestures. Research by Cadoz and Wanderley [3] has stressed the importance of gesture classification and recognition. Previous research by Cadoz [2] also emphasized the importance of haptic feedback for the design of interactive interface for sound production: the physical feedback given by the intermediary device - such as a Wii remote in our case - contributes to create memorizable gestures, and complete the audio feedback rendered by the interface. Kela et al. [5] studied the use of accelerometers for multi modal activities; applications in music have also been studied (see e.g. [8]). However, the algorithms presented for this research can be used with other gesture controllers, such as motion capture or any sensor-based technology. Whilst other approaches have focussed on pre-defined classification, we are interested in real-time classification for use in music performances. A starting point of our research was to develop an on-the-fly learning of specific gestures, in order to create a database of recognizable gestures that could be shared between performers. The first part of this paper describes two algorithms used to recognize a fixed length gesture. The second part presents a dynamic and unsupervised recognition model that is able to handle various length gestures. The two methods are discussed and future works are presented.

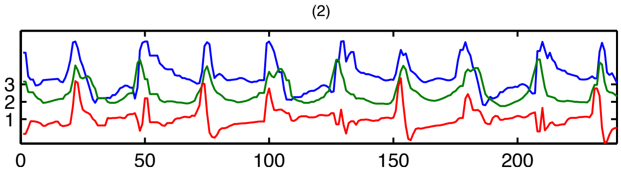

Figure 1: The three accelerometer signals captured from a Wii controller while making repeated gestures.

2. SUPERVISED METHOD WITH HAPTIC FEEDBACK

The Wii remote controller is a popular and pervasive device that detects 3-dimensional movements via three accelerometers, one for each dimension (relative to the controller). The signals produced by the accelerometers are transmitted via Bluetooth to a laptop computer. We used an external object within Max/MSP developed by Masayuki Akamatsu to decode the transmissions from the controller. The three signals sent by the controller are sampled at rate of 50 Hz with an accuracy determined by the Max/MSP internal timing system. The latency produced by bluetooth devices has been estimated to approximately 50ms [9]. However, more precise measures of both latency and sampling jitter still need to be made. The Fig. 1 shows an example of how the data evolves over a fixed period of time.

The Wii device can produce a vibration that we use as feedback to the user when a gesture is recognized. In addition, a visual cue is produced. We now turn to a method implemented in order to categorize a gesture in real time with supervision.

2.1 First method: Euclidean distance

In this method, a window of controller signals is stored. The length of the window is determined by the duration of the gesture to be recognised, so that the length in samples,

L, will depend on the sampling rate, e.g. 6 (x, y, z) triplets at 50 Hz for a gesture that lasts 120ms. The user triggers the capture of a template or reference gesture by pressing a button ('A') on the controller at the *end* of the movement[1]. At this stage, the system is ready to compare fragments of the incoming data with the reference gesture.

If the reference gesture V_r is considered as a $3L$-dimensional vector, and V_i is a similar vector constructed from the last L samples of the input signal, then the Euclidean distance between the reference and the input is

$$D = \sqrt{(V_i - V_r) \cdot (V_i - V_r)}, \tag{1}$$

where for our purposes the dot product is defined as

$$A \cdot B = \sum_{i=1}^{L} A_x(i)B_x(i) + A_y(i)B_y(i) + A_z(i)B_z(i), \tag{2}$$

that is, a sum over the L samples and the three dimensions. The gesture is detected when the distance drops below, or reaches a minimum below, a given threshold, as shown in fig. 2(b).

2.2 Second method: cosine similarity

The cosine of the angle between the reference vector and the input vector can be computed by taking the dot product and dividing by the norms of the two vectors:

$$C = \frac{V_r \cdot V_i}{\sqrt{V_r \cdot V_r}\sqrt{V_i \cdot V_i}}, \tag{3}$$

using the same definition of the dot product as before. It is 1 when the vectors are parallel, i.e. the gestures are identical up to an arbitrary scaling factor. Thus, we can detect gestures similar to the reference by looking for peaks in the cosine above a certain threshold, as shown in fig. 2(c).

2.3 Discussion

Supervised recognition, in both cases presented above, seems to be an appropriate method for the definition of precise gestures. By focusing on one gesture at a time, we are able to repeat a movement several times until the vibration produced (as a result of the recognition) arrives at the moment it is expected. Moreover, the issue of latency due to the various processing steps can be addressed. A gesture can be recognized before it is finished as long as its initial fragment can reliably be recognized in advance. In our case, we observed that initial fragments of more than 80ms are usually distinct enough not to be confused with other gestures. If we increase the 'anticipatory lag' by choosing a gesture template from an initial fragment that ends well before the end of the gesture, the haptic feedback can be triggered at the time the performer expects, but on the other hand, the detection is less reliable. The number of entries of the constituted database is also an important factor in the overall error rate.

We chose to analyse a regular, repeated movement, consisting of a cycling through three hand movements, visible as the large peaks in fig. 2(a). One of these movements, extracted from near the beginning of the signal, was taken to be the reference gesture—it is visible in fig. 2(b) at the point where the Euclidean distance drops to zero. As it is shown in figure 2, repetitions of the same gestures are not identical, therefore the threshold for detection must be larger than 0 or less than 1 for the two methods respectively. The cosine method, being invariant to overall magnitude of the accelerometer signals, is able to recognize the reference gesture even if it is performed at a larger scale, as long as it has the same duration.

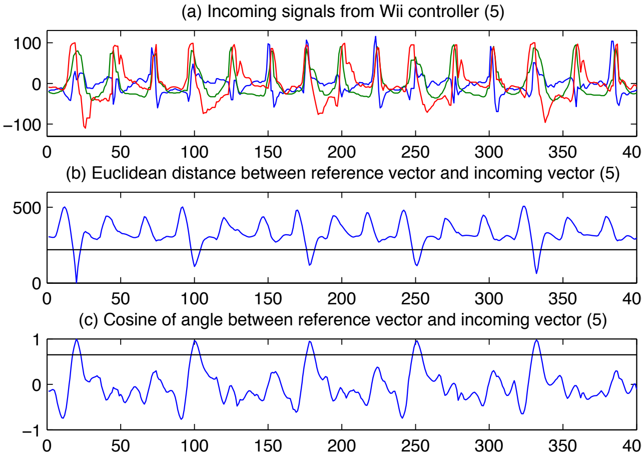

Figure 2: Analysis of the signal for a repeated gesture. The reference gesture was taken from the begining and is visible where the distance drops to zero. Suitable thresholds for detection are shown as black horizontal lines.

Both methods are quite sensitive to the choice of reference gesture and the thresholds, but in this case we were able to find parameters that gave successful detection of all 45 instances of the reference gesture using the cosine methods, and 44/45 using the Euclidean distance measure, with no false positives. We were also able to use the results of the initial run to construct a better reference gesture by averaging all the previously detected instances. This gave perfect results using both methods.

3. UNSUPERVISED METHOD USING INFORMATION DYNAMICS

The above supervised method requires two distinct pieces of information to recognise a gesture in a timely way: one is the reference gesture with its label and the other is the indication of the particular time point, relative to the reference, at which to respond to the gesture. This can be thought of as a mark indicating the 'perceptual centre' of the gesture (see fig. 3).

Though in some applications it may be possible to interleave the training phases with the performances phases, as we did in the system described above, in other applications it may not be possible for the person or system creating the gestures to provide this extra stream of information stating that '*this* is gesture *A*', '*this* is gesture *B*', and so on. For example, a dancer's movements might be improvised and the dancer too occupied with the actual execution of them to be able to mark and label them as well. However, human observers are capable of recognising a repeated gesture and inferring a series of relatively precise timings from what is on the face of an unstructured continuous movement.

[1] Pressing the button while doing the gesture is not an appropriate solution in the long term, as it affects the gesture itself. This problem is addressed in the unsupervised version (see section3).

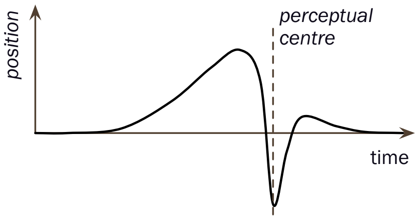

Figure 3: A one dimensional gesture (e.g. a hand moving up and down) where the implied punctual event or beat is marked as the perceptual onset and is some time after the initial onset.

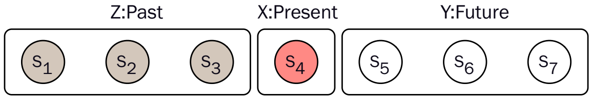

Figure 4: Representation of a sequential perceptual process: at any given time, there will be context of known previous observations, a 'current' observation, and an unobserved future.

3.1 Predictive information

The question at the heart of gesture recognition is how do we perceive discrete and punctual (that is, associated with a particular *point* in time) events in a continuous signal? Our approach to this is to consider the *predictive information rate* of the signal as processed by the observer. Essentially, we consider our hypothetical observer to be engaged in a continuous (and largely unconscious) process of trying to predict the future evolution of a signal as it unfolds. These predictions are *probabilistic* in nature; that is, they entail the assignments of probabilities to the various possible future developments.

A sufficiently adaptive perceptual system will internalise any statistical regularities in the signal, such as smoothness or any typical or repeated behaviour, in order to make better predictions. If a particular observation, which in practical terms might consist of a few samples of motion capture data, brings about a large change in the predictive probability distribution, then we associate with it a large *predictive information*. In this way, we can plot the predictive information rate against time. Referring to fig. 4, the predictive information is the Kullback-Leibler divergence (a measure of distance between probability distributions) between $P(Y|Z=z)$ and $P(Y|Z=z, X=x)$, where $Z=z$ and $X=x$ denote the propositions that past and present variables respectively were observed to have particular values z and x.

Now, depending on both the signal *and* the observer's predictive model, the predictive information rate can take many forms, but in particular, it may in some cases be relatively flat, while in others, more peaky or bursty, in the sense that the predictive information arrives in concentrated 'packets' interspersed by longer periods of relatively low predictive information. It is in this latter case that we identify the 'packets' of information as the 'events'.

3.2 HMM-based implementation

We have implemented a version of this hypothetical observer using a relatively simple predictive model (a Markov chain) in which the predictive information associated with each observation can be computed quite straightforwardly[2].

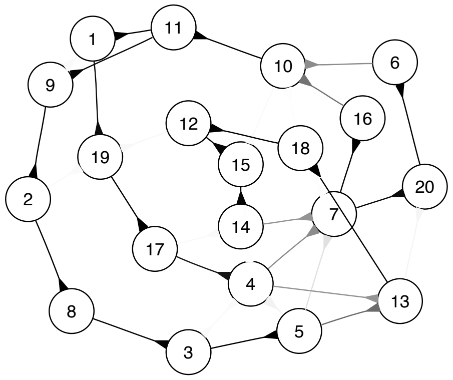

Figure 5: The state space of one of the HMMs trained on the recorded data. The directed edges represent the transitions; self-transitions and transitions with very low probability have been hidden. The darkness of the edges shows the probability of the corresponding transition.

The analysis proceeds as follows. The three sampled signals are windowed, taking L consecutive samples, and represented as a vector with $N = 3L$ components. At each time step the window is shifted along by one sample. The resulting sequence of vectors is taken as the as the continuous-valued observation sequence from a hidden Markov model (HMM) with Gaussian state-conditional distributions and K possible states. The parameters of this HMM (the transition matrix and the mean and covariance for each of the K states) are trained using a variant of the Baum-Welch algorithm [7]. Once the HMM is trained, the most likely sequence of hidden states is inferred using the Viterbi algorithm and the information dynamic analysis applied to the Markov chain.

Many instances of the system were trained with different random initialisations. Fig. 5 shows the underlying Markov chain found in one such instance with $L = 9$ and $K = 20$. The transition structure shows that there a small number of typical paths through the state space, corresponding to different gestures. Our information dynamic analysis automatically picks out states which most effectively signal that a particular path is being traversed; in the figure, the most informative states are 17, 8, and 15. Note that state 3 is not as informative as state 8 as state 3 has a high self transition probability.

In fig. 6, the variation in predictive information rate over time is shown (this example actually uses a different HMM from that shown in fig. 5). Event detection then proceeds by picking all transitions with a predictive information greater than a fixed threshold, and the identity the target state is used to categorise the event. In our experiments, we sonified these events using a different pitch for each event type. In most cases, all the gestural events (approximately 150 in total) are detected and categorised into 2–4 classes, with 1–3 false positives.

[2] However, the Markov chain is not observed but inferred using a hidden Markov model (HMM) so there is an element of approximation involved

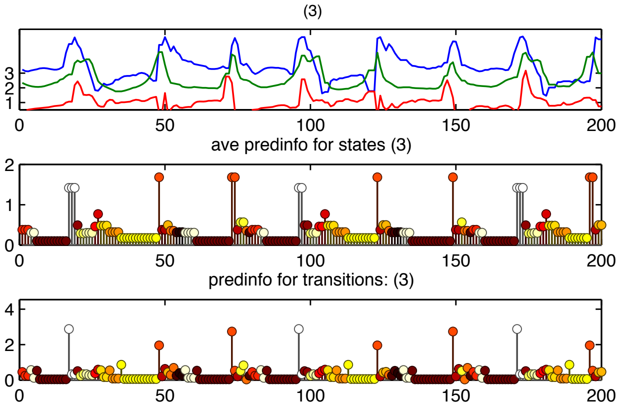

Figure 6: Information dynamic analysis of accelerometer signals in top panel. The middle panel shows the state sequence inferred from the HMM in a way that highlights the average informativeness of each state in the sequence: the shading of each marker encodes which of the 20 states is active, while the y-axis represents the average predictive information associated with that state. In the bottom panel, the shading encodes the state as before, but the y-axis encodes the predictive information associated with that particular *transition* in context.

3.3 Related work

Hidden Markov models have been applied to gesture recogniton by many researchers [4, 6]. In the terminology used in this field, our system performs *continuous* gesture recognition because there are no given boundaries between gestures. Our current HMM based system is not *online* but could easily be made so using fixed-lag decoding of the HMM instead of the current off-line Viterbi algorithm.

Unlike other HMM-based systems of which we are aware, our system uses a *single* HMM to model all gestures instead a separate HMM for each one. Thus the categorisation of input signals as one gesture or another is made through the normal operation of the forwards-backwards or Viterbi algorithms.

In fact our system is more closely related to the audio onset detection system described in [1]. The difference is that in the earlier system, the choice of which states were to be taken as indicators of significant events had to made manually, where as the current system uses information dynamic principles to do this automatically.

4. CONCLUSION

In this work, we have investigated the development of efficient tools for real-time gesture recognition. The Nintendo Wii remote was chosen to provide data to our methods, however, both supervised and unsupervised algorithms are adaptive enough to deal with signals from different controllers. The template matching system is based on well-known template matching methods, while the HMM based system uses novel information-theoretic criteria to enable unsupervised identification of an initially unknown number of gestures. At this stage, the recognition part of the HMM-method is implemented in Matlab, but could be implemented in real-time fairly straightforwardly using a standard fixed-lag smoothing algorithm for the HMM [7]. The explicit probabilistic formulation of the model makes it well suited to handling the detection latency problem by predicting the future motion of the controller *and* estimating how accurate this prediction might be. The supervised method, however, is implemented in Java as a plug-in for Max/MSP and works in real-time. An external to calculate the Euclidean and cosine matching methods for any signal will be soon be released. Online training of HMMs is possible but is an inherently more difficult problem which we are researching currently.

Part of the motivation behind this work is that multiple performers could use the system and thereby share information about gestures made. For example, when a gesture triggers or schedules a sonic or visual event, it could also cause a vibration signal to be sent to the other performers' controllers. This extra level of *haptic* communication could enhance the sonic and visual interaction without interfering with the performance as seen and heard by the audience. Future work will explore the importance of shared cues between performers and the development of haptic solutions to communicate these cues.

5. ACKNOWLEDGMENTS

This work was partly supported by two EPSRC grants: GR/S82213/01 and EP/E045235/1. A. Robertson and J.-B. Thiebaut are supported by EPSRC Research studentships.

6. REFERENCES

[1] S. Abdallah and M. Plumbley. Unsupervised onset detection: a probabilistic approach using ICA and a hidden Markov classifier. In *Cambridge Music Processing Colloquium*, Cambridge, UK, 2003.

[2] C. Cadoz, A. Luciani, J. Florens, C. Roads, and F. Chadabe. Responsive input devices and sound synthesis by stimulation of instrumental mechanisms: The cordis system. *Computer Music Journal*, 1984.

[3] C. Cadoz and M. Wanderley. Gesture - music. In M.M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. Ircam - Centre Pompidou, 2000.

[4] Y. Iwai, H. Shimizu, and M. Yachida. Real-time context-based gesture recognition using hmm and automaton. In *Proc. of the Int. Workshop RATFG-RTS '99*, Washington, 1999. IEEE Computer Society.

[5] J. Kela, P. Korpipaa, J. Mantyjarvi, S. Kallio, G. Savino, L. Jozzo, and S. Di Marca. Accelerometer-based gesture control for a design environment. *Personal and Ubiquitous Computing*, pages 285–299, 2006.

[6] C. Lee and Y. Xu. Online, interactive learning of gestures for human/robot interfaces. In *IEEE Int. Conf. on Robotics and Automation*, pages 2982–2987, 1996.

[7] L. R. Rabiner. A tutorial on hidden markov models and selection applications in speech recognition. *Proc. of the IEEE*, 77(2):257–286, 1989.

[8] H. Sawada and S. Hashimoto. Gesture recognition using an accelerometer sensor and its application to musical performance control. *Electron Commun Jpn*, Part 3:9–17, 2000.

[9] Sena Technologies. White paper: Latency/throughout test of device servers/bluetooth-serial adapters. Technical report, Sena Technologies, 2007.