A Primitive Based Generative Model to Infer Timing Information in Unpartitioned Handwriting Data

Ben H Williams, Marc Toussaint and Amos J Storkey
Edinburgh University
School of Informatics
ben.williams@ed.ac.uk

Abstract

Biological movement control and planning is based upon motor primitives. In our approach, we presume that each motor primitive takes responsibility for controlling a small sub-block of motion, containing coherent muscle activation outputs. A central timing controller cues these subroutines of movement, creating complete movement strategies that are built up by overlaying primitives, thus creating synergies of muscle activation. This partitioning allows the movement to be defined by a sparse code representing the timing of primitive activations. This paper shows that it is possible to use a factorial hidden Markov model to infer primitives in handwriting data. The variation in the handwriting data can to a large extent be explained by timing variation in the triggering of the primitives. Once an appropriate set of primitives has been inferred, the characters can be represented as a set of timings of primitive activations, along with variances, giving a very compact representation of the character. The model is naturally partitioned into a low level primitive output stage, and a top-down primitive timing stage. This partitioning gives us an insight into behaviours such as scribbling, and what is learnt in order to write a new character.

1 Introduction

Biological systems are strongly superior to current robotic movement control systems, despite having very noisy sensors, and unpredictable muscles. Therefore, the amount and nature of pre-planning in biological movement is extremely interesting. Strong evidence exists to suggest that biological motor control systems are modularised, with *motor primitives* first being conclusively found in frogs [Bizzi *et al.*, 1995; d'Avella and Bizzi, 2005; d'Avella *et al.*, 2003], where stimulation of a single spinal motor afferent triggered a complete sweeping movement of the frog's leg. For a review of modularisation of motor control in the spine, see [Bizzi *et al.*, 2002].

Evidence suggests that once a particular subsection of movement has commenced, it cannot be unexpectedly switched off. Rather, to quickly modify a movement trajectory, the movement primitives are superimposed [Kargo and Giszter, 2000].

A primitive in the handwriting domain will be a short time extended block of pen motion, brought about by a corresponding block of muscle activations. These blocks are superimposed upon each other to create coherent movement - in this case, a character.

To reliably model natural handwriting, we introduce a fully generative model, allowing proper modelling of handwriting variation and adaptation irrespective of the character drawn. If the handwriting output is made up of primitive type sub-blocks then the generative model must represent these primitives, to allow it to efficiently model the internal handwriting encoding.

In section 2, we introduce our generative handwriting model based on primitives in terms of a factorial hidden Markov model. Section 3 covers the generalisation over the timing of primitives to create a timing model. In section 4 we present some typical samples, and the results of using this model, and finally the discussion is in section 5.

2 Model

A generative handwriting model must be capable of reproducing the class of characters upon which it has been trained. Assuming that all motor commands are made up of motor primitives, handwriting must therefore contain projections of these primitives. Assuming also that motor primitives are fixed, or adaptable over long time scales, any short term adaptability and learning must come from the timing and selection of different primitives.

Assuming the individual primitives are independent of each other, and are linearly superimposed, a controller to select primitive onset timing is necessary, similar in nature to a Piano Model, where key pressing controls the onset of time extended clips of sound that the listener hears as a superimposed waveform.

2.1 A Deterministic Piano Model

To formalise this model in a generative way, the output of the system Y at time t is defined as

$$Y(t) = \sum_{m,n} \alpha_{mn} W_m(t - \tau_{mn}) \,, \tag{1}$$

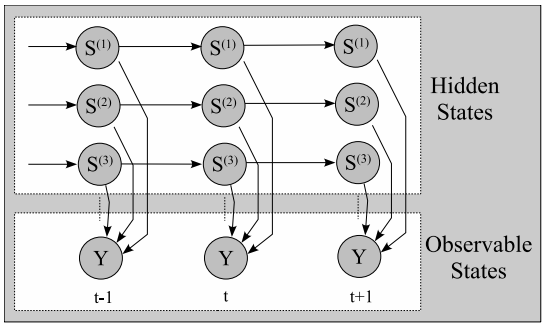


Figure 1: Graphical representation of a Factorial Hidden Markov Model, showing the independence of the separate Markov chains. Although the observable output is dependent upon the state of the entire system, the internal states evolve with no interdependencies. S_t^m denotes the hidden state vector at time t, in factor m.

where $W_m(t)$ are the primitives, and τ_{mn} represents the time of the n^{th} activation of primitive m, and α_{mn} defines the activation strengths of the primitives. In this definition, m enumerates the primitives, whilst n enumerates the occurrence of the primitive within the sample window, at time τ_{mn}.

Similar models have been used for modelling real piano operation such as [Cemgil *et al.*, 2005], where the transfer from piano roll to sound uses note onset, and damping onset as key timing information for the reproduction of sound. Also [Karklin and Lewicki, 2005] present a generative model of speech waveforms, where their 'Sparse Shiftable Kernel Representation' is similar in nature to the Piano Model presented here.

The data samples of handwritten characters are not segmented or keyed to a common start point, apart from the pen touching the paper. As this is not a reliable keying time for primitive activation, a flexible model must be used to infer the primitives, which will not only infer the shape of the primitives, but their timing onsets. We take the idea of the Piano Model as a basis, but model it probabilistically using a factorial hidden Markov model (fHMM).

2.2 Factorial Hidden Markov Model

A graphical model of the fHMM can be seen in Figure 1. At each time step, the observable output Y_t, a vector of dimension D, is dependent on M hidden variables $S_t^{(1)}, .., S_t^m$. The output is a multivariate Gaussian, such that

$$Y_t \sim \mathcal{N}(\mu_t, C) \,, \tag{2}$$

where C is a $D \times D$ parameter matrix of output covariance, and

$$\mu_t = \sum_{m=1}^{M} W^m S_t^m \tag{3}$$

is the D-dimensional output mean at time t. W^m is a $D \times K$ parameter matrix giving the output means for each factor m, such that the output mean μ_t is a linear combination of its columns weighted with the hidden state activations.

Each of the M hidden variables can be in K different states. In equation (3) this is encoded in the K-dimensional state vector S_t^m using a 1-in-K code, i.e., $S_{t,i}^m = 1$ if the m-th factor is in state i and zero otherwise. This allows us to write expectations of the hidden states as $\langle S_t^m \rangle$, which is also the probability distribution over the individual states S_t^m. Each latent factor is a Markov chain defined by the state transition probabilities and the initial state distribution as

$$P(S_1^m = i) = \pi_i^m \,, \qquad P(S_t^m = i \,|\, S_{t-1}^m = j) = P_{i,j}^m \,, \tag{4}$$

where π^m is a K-dimensional parameter vector giving the initial hidden state distribution, and P^m is a $K \times K$ parameter matrix denoting the state transition probabilities. As can be seen in Figure 1, each factor is independent. This means that the joint probability distribution can be factorised as

$$P(\{Y_t, S_t\}) = P(S_1)P(Y_1|S_1) \prod_{t=2}^{T} P(S_t|S_{t-1})P(Y_t|S_t) \tag{5}$$

$$= \prod_{m=1}^{M} \pi^m P(Y_1|S_1) \prod_{t=2}^{T} \prod_{m=1}^{M} P^m P(Y_t|S_t) \,. \tag{6}$$

Given the fully parameterised modelling framework, learning of the parameters can be done using an Expectation-Maximisation (EM) method. The structured variational approximation was chosen for the E-step inference. For more details on the various arguments for and against this choice, refer to [Ghahramani and Jordan, 1997], which provides details about the fHMM model, and the learning procedure.

The EM method is an iterative algorithm, in which the E-step infers the expectations of the hidden states given a set of parameters, then the M-step updates the parameters to their maximum-likelihood values, given the inferred hidden state distributions. In our case, the E-step fits the primitives to the data, inferring the primitive onset timings for each sample. The M-step infers the shapes of the primitives. Some constraints were imposed upon the parameters, so that the primitives progressed monotonically along their states, and that the rest state for all primitives, gave a zero output contribution.

The fHMM can reconstruct the data by using a set of maximally statistically independent primitives, and the appropriate hidden state values. Due to the constraints imposed upon the hidden state transitions, these state values can be reduced to a set of primitive activation timings, or *spikes*. Without this spike timing information, the primitive model can still be run separately, as can be seen in Figure 4, which can be thought of as *primitive babbling*. To reconstruct a character, the primitives need to be coordinated, and activated at the appropriate times. This is achieved by introducing a separate part of the model, the centralised timing controller.

3 Timing Model

The centralised timing controller must be capable of reproducing spiking characteristics that in some areas of the character are variable, and others less so, both in time of spike, and existence of spike. In other words, some primitives are necessary, occurring in every character sample in roughly, but not exactly the same place. Others occur occasionally in a reproduction, but not in every case. Crucially, on a short time

scale, there is heavy dependency between spikes, whereas on the long term, they are simply dependent upon the character being drawn.

We have chosen a stochastic Integrate and Fire (IF) model for the generation of spikes, as this model has a local temporal dependency characteristic, and also allows variance in the total number of spikes in a sample.

Integrate and Fire The Integrate and Fire (IF) model originates from simplified models of biological neurons. It treats the neuron as a leaky capacitor, upon which a charge is built up by the inputs, over time. Once the voltage across the capacitor reaches a noisy threshold level, the neuron fires, producing a spike at its output, and discharging the capacitor. This means that, due to the leak term, over a long time scale, the inputs at different times are independent, however, on a short time scale, they are not, as it is the short-term running sum of the inputs that causes the neuron to fire. This is desirable for the primitive model, because the timing of a necessary primitive can be variable in the character samples, however, the IF neuron will still fire as long as it receives enough inputs during its temporal memory window.

The most straight forward model using IF neurons is to attribute one IF neuron to one primitive. The inputs to the neurons will then determine the timing of the primitives and thus the character that is written. For a particular primitive, m, the probability of a spike at time t, $P(\lambda_t^m)$ is given by:

$$P(\lambda_t^m|\lambda_{t-1}^m = 0) = P(\lambda_{t-1}^m) + I_t^m - L_t^m, \quad (7)$$

$$P(\lambda_t^m|\lambda_{t-1}^m = 1) = I_t^m - L_t^m, \quad (8)$$

$$L_t^m = \nu P(\lambda_{t-1}^m), \quad (9)$$

where I_t^m are the input excitations, and L^m is a leak term proportional to the accumulated probability. Therefore, given a common set of primitives, a character is defined by its *temporal excitation matrix*, I_t^m, which parameterises the IF model. This matrix is learnt from the spiking statistics of the character training set, as seen below.

During the E-step for the fHMM, the hidden state distribution $P(S_t^m)$ is inferred. As the transition matrix is constrained so that the primitives progress monotonically through their states, the information in $P(S_t^m)$ can be summarised as the onset probabilities of the different primitives, $P(S_{t,1}^m = 1)$ which are the rows of $P(S_t^m)$ for state 1, the first state in each primitive. For a set of primitives that fit the data well, these probabilities are close to zero or one, and form a very sparse matrix containing spikes representing primitive activation appropriate to reconstruct a single character sample from the data set. It is effectively an average over the samples of these spiking matrices that is needed to parameterise the IF model, as I_t^m. For ease of notation, let $\tau_{t,n}^m = P(S_{t,1}^m = 1)$ be the posterior onset probability at time t of the mth primitive for the n data sample.

To allow for differences in the start point time, and average speed of each character sample, two parameters are associated with each data sample, a temporal offset δt_n and a linear temporal stretching factor δl_n. These parameters are optimised so that the $\tau_{t,n}^m$ matrices for the nth sample best fit the average I_t^m matrix that is constructed by taking linear interpolations.

$$I_t^m = \frac{\sum_n \tau_{k_n,n}^m}{N}, \quad k_n = (t + \delta t_n)\delta l_n. \quad (10)$$

More precisely, we optimize the temporal offset and strechting by iteratively finding δt_n and δl_n via gradient ascent that maximize the objective function

$$\sum_{n,m,t} \tau_{k^n,n}^m I_t^m \,. \quad (11)$$

This finds an I_t^m matrix that best reflects an average primitive onset probability matrix, where t has a separate linear shifting and stretching factor associated with it for each character sample, n. This is used to parameterise the IF model, which generates the spike timing information needed to run the fHMM generatively.

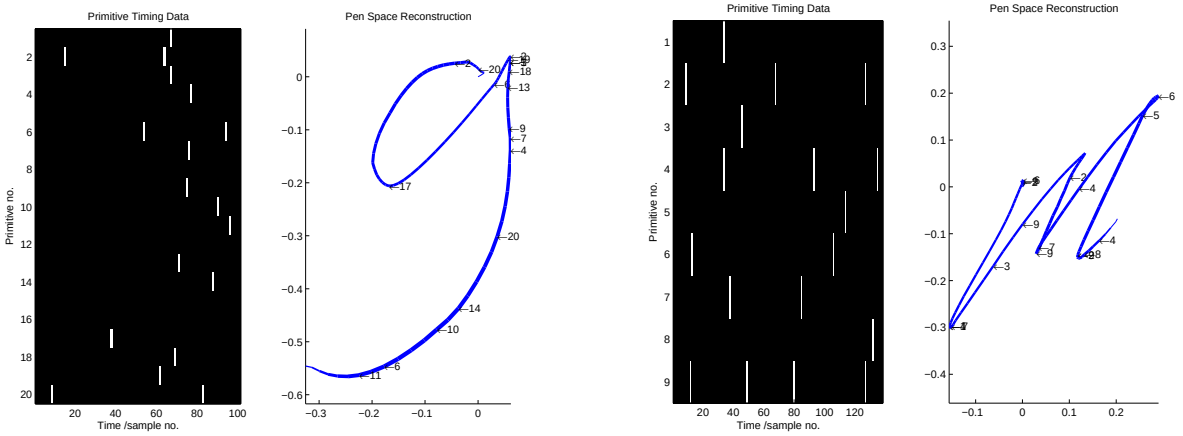
3.1 Implementation

Handwriting data were gathered using an INTUOS 3 WACOM digitisation tablet `http://www.wacom.com/productinfo/9x12.cfm`. This provided 5 dimensional data at 200Hz. The dimensions of the data were x-position, y-position, pen tip pressure, pen tilt angle, and pen orientation (0-360°). The normalised first differential of the data was used, so that the data mean was close to zero, providing the requirements for the zero state assumption in the model constraints (see section 2.2). Only 3 dimensions of the data were used, x-position, y-position, and pressure, as the signal to noise ratio of the other two was too low to provide usful data. The data collected were separated into samples, or characters, for processing purposes, and then the parameters were fitted to the data using our algorithm. We generated datasets of the characters *'g'*, *'m'*, *'a'*, *'b'*, *'c'* and *'d'*. The *'g'* character set was the largest character set, and can be found at `http://homepages.inf.ed.ac.uk/s0349619/data/mixoutG.mat`. The character sets *'g'* and *'m'* were chosen to explore the differences in primitives inferred from two contrasting characters, see Figure 4. The characters *'a'*, *'b'*, *'c'* and *'d'* were chosen to explore the sharing of primitives between different characters, and the ability to create distinct reconstructions, as seen in Figure 7.

Data set	Size
'g'	1292
'm'	125
'abcd'	296

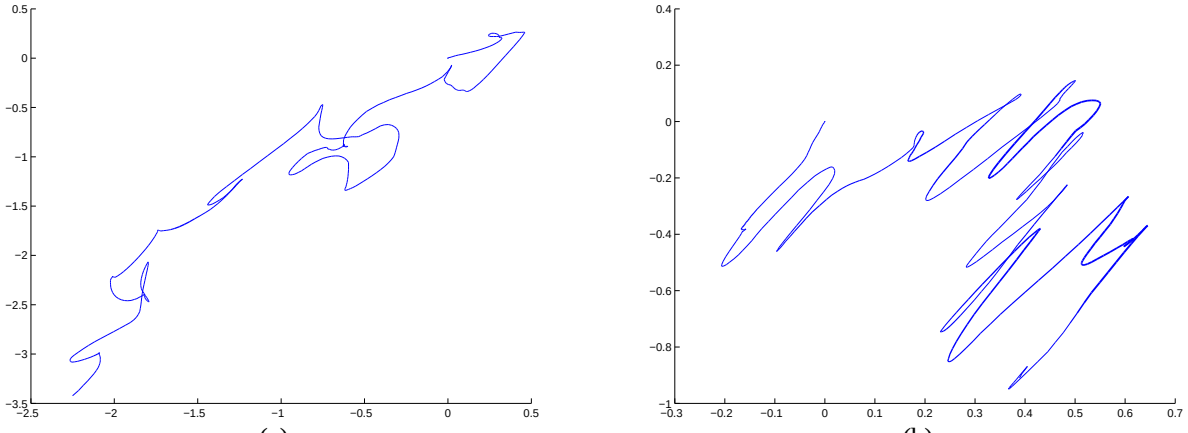
4 Results

A typical subset of primitives learnt via the fHMM from data of g-characters are shown in Figure 2 demonstrating the variation in shape and length on the paper of the primitives when reconstructed on their own. In this example, 20 primitives of length 40 were used to model a data set of 1292 characters, of average length 103.8. The average error of the reconstruction in velocity space for the data was $[0.0617\ 0.0601\ 0.0295]$. Using these primitives, the fHMM model can reconstruct the original character as shown in Figure 3 using the posterior timings $\langle S_t \rangle$ that are inferred from the data in the E-step of



(a) (b)

Figure 3: Two examples of character reconstructions, using timing information derived from the E-step inference. In both (a) and (b), the timing information is shown on the left, and the reproduction of the sample on the right, with the onset of each primitive marked with an arrow. In (a), 20 primitives of length 40 time steps were used to model the *'g'* dataset, of over 1000 characters. In (b), 9 primitives of length 30 were used to model the *'m'* dataset, of over 100 characters.



(a) (b)

Figure 4: Two samples generated using primitives without specific timing information (neither the posterior timing nor a learnt timing model). (a) was generated using primitives inferred from the *'g'* dataset, (b) was generated using primitives from the *'m'* dataset. Starting point of both reconstructions is $(0, 0)$.

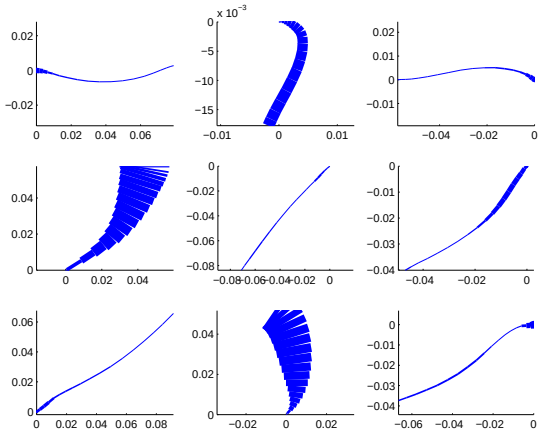


Figure 2: A sample of 9 primitives used for the reconstruction in Figure 3(a). The axis values refer to some normalised distance on paper, as the primitives are shown as a reconstruction in pen-space, rather than the raw data, which is in 3-dimensional velocity space. As the data was normalised ignoring the physical size of the pixels, it would be meaningless to provide units for the axes. Thickness corresponds to pen pressure. Refer to section 3.1 for an explanation.

the primitive extraction stage. The timing information, seen on the left of the Figures is a representation of the posterior probabilities of onset of the primitives whilst reconstructing the data set. These probabilities are inferred during the E-step, and reveal the expected hidden state values of the fHMM. Furthermore, given a sufficient set of common primitives to model the data reliably, these probabilities can be represented as spike timings, which provide a very compact encoding of the character. This timing information can also be modelled using a timing model, as investigated in Section 3. Here, no timing model was used or learnt. Without such a timing model and without the hidden state posterior extracted from the data, one can still run the primitive model generatively, ignoring any precise timing information, and sampling the model states from the priors, without calculating the posterior distributions. The result is a random superposition of primitives producing a scribbling form of output such as shown in Figure 4. Here it can be seen that the primitives, even without precise timing information controlling their activation, still generate samples that capture an aspect of the training set from which they were extracted. Allowing the temporal shifting and stretching as described in Section 3, produces a distribution over spiking patterns, I_t^m as can be seen in Figure 5 for the *'g'*-dataset. Sampling from this distribution using our *integrate and fire* approach as described above, produces samples that reliably model the variation of the data set, as can be seen in Figure 6. Clearly, these are all examples of a character *'g'*, however, the pen trajectory is vastly different in each example, reflecting the variance in the original data set. Inferring primitives from a range of characters is also possible. Figure 7 shows 4 different characters all drawn using the same primitives. Despite the variation in the characters of a particular alphabet, the actual hand movements when controlling the pen are very similar, so it should come as no surprise that it is possible to use the same primi-

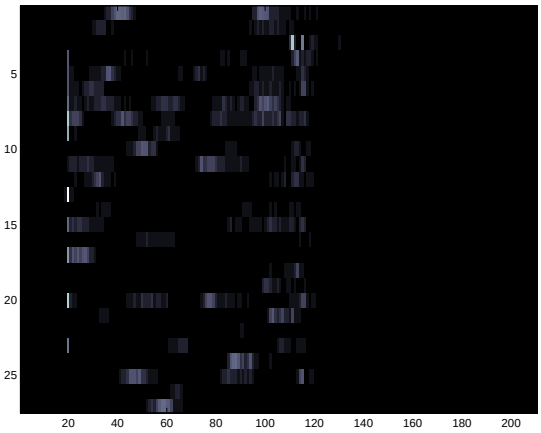


Figure 5: A graphical representation of the distribution of spikes as given by I_t^m for a *'g'* character set.

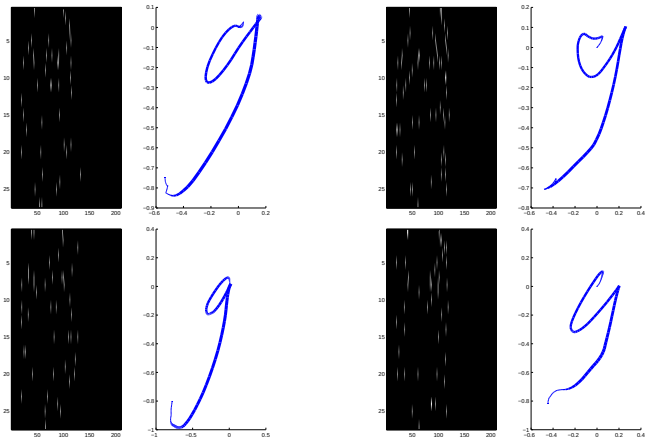


Figure 6: Four samples generated from the full generative model as given by the learnt primitives (the fHMM) and the timing model parameterized by the spike distribution I_t^m (shown in Figure 5).

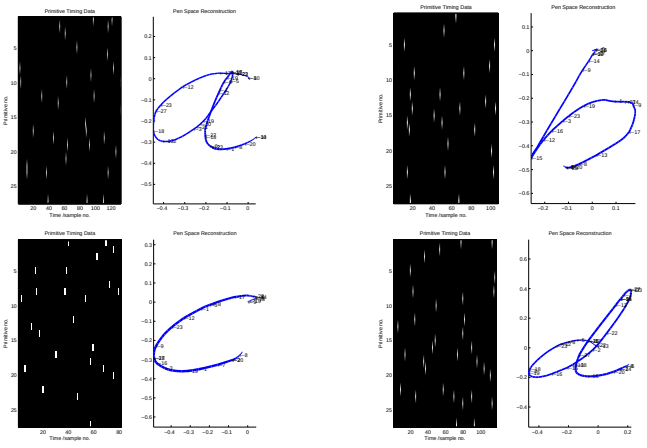


Figure 7: Four characters that have been reproduced using the same set of primitives, learnt from a mixed character set.

tives to reconstruct different characters. The variation in the characters must therefore be accounted for by the different timings of the primitives.

5 Discussion

Using the fHMM framework to infer motor primitive representations in handwriting data gives us primitives that can be active at any point, that are statistically independent, and are not limited to any particular section of character, or any subset of characters.

The spike timing distribution that can be seen in Figure 5 may also be expressed in terms of a mixture of Gaussians, where the mean of a Gaussian conveys the average timing of a particular spike that is necessary to reproduce a character. This would be a more compact encoding for a character, although it would restrict the spike timing model to have the same number of spikes for a particular primitive in all samples. The stochastic Integrate and Fire model allows variance in the number of spikes, and includes the short term time dependency that is necessary to produce meaningful spike timings, however, the parameterisation matrix is much larger than would be the case for a mixture of Gaussians.

Without the timing model, the primitives can still produce output, by running the fHMM model generatively, and sampling the hidden states at each time step, from the prior distribution conditioned on the previous time step. This produces an output sample that is similar to scribbling or doodling. Perhaps when we are absent-mindedly doodling, it is a disconnection of the timing control part of our motor system from the muscle output, or primitive section that produces these scribbles. Intuitively, we are putting a pen in our hand, and telling ourselves to write, but not dictating what exactly.

In this model, given a common set of primitives used to control the complete set of possible movements, it is clearly the timing model that is dictating what class of movement is required. The spikes are the internal encoding of the movement class. Using the spike timing representation of when the primitives are triggered, allows a very compact representation for a character. As a common set of primitives is capable of reconstructing several different characters, the spike encoding can therefore be divisive in differentiating one character from another, and may therefore be useful for efficient character recognition. The compact code would also be useful for data storage and transmission, in much the same way as the Ascii code allows efficient transmission of printed characters, a spike encoding could efficiently transmit handwritten ones.

This distributed model provides a framework both for learning new character sets, and experimentation with primitive shapes. It is possible that primitive shape is a personal attribute, which may account for different handwriting styles. If this is so, then an efficient mapping from one style to another may be possible. This hypothesis requires further research, by examining the primitives inferred from different people's handwriting samples. Another hypothesis may be that different people have a different number or average length of primitive, maybe this could account for more 'messy' styles of handwriting.

Given a common set of primitives, the spike timings encode the movement, but this does not address learning new movements. When we learn to write for instance, we must either learn new primitives, or adapt old ones, or perhaps some movements are simply not learnable or rather cannot be encoded by a sparse spike representation. The adaptation of primitives, and their association with learning new motor skills would be a very interesting area to explore, with the help of this model.

References

[Bizzi *et al.*, 1995] E. Bizzi, S.F. Giszter, E. Loeb, F.A. Mussa-Ivaldi, and P. Saltiel. Modular organization of motor behavior in the frog's spinal cord. *Trends in Neurosciences*, 18(10):442–446, 1995.

[Bizzi *et al.*, 2002] E. Bizzi, A. d'Avella, P. Saltiel, and M. Trensch. Modular organization of spinal motor systems. *The Neuroscientist*, 8(5):437–442, 2002.

[Cemgil *et al.*, 2005] A. Cemgil, B. Kappen, and D. Barber. A generative model for music transcription. In *IEEE Transactions on Speech and Audio Processing*, volume 13, 2005.

[d'Avella and Bizzi, 2005] A. d'Avella and E. Bizzi. Shared and specific muscle synergies in natural motor behaviors. *PNAS*, 102(8):3076–3081, 2005.

[d'Avella *et al.*, 2003] A. d'Avella, P. Saltiel, and E. Bizzi. Combinations of muscle synergies in the construction of a natural motor behavior. *Nature Neuroscience*, 6(3):300–308, 2003.

[Ghahramani and Jordan, 1997] Z. Ghahramani and M.I. Jordan. Factorial hidden Markov models. *Machine Learning*, 29:245–275, 1997.

[Kargo and Giszter, 2000] W.J. Kargo and S.F. Giszter. Rapid corrections of aimed movements by combination of force-field primitives. *J. Neurosci.*, 20:409–426, 2000.

[Karklin and Lewicki, 2005] Y. Karklin and M. S. Lewicki. A hierarchical bayesian model for learning non-linear statistical regularities in non-stationary natural signals. *Neural Computation*, 17(2):397–423, 2005.