myjournal manuscript No.
(will be inserted by the editor)

Analyzing Movement Trajectories Using a Markov Bi-Clustering Method

Keren Erez[1] · **Jacob Goldberger**[2] · **Ronen Sosnik**[1] · **Moshe Shemesh**[1] · **Susan Rothstein**[1] · **Moshe Abeles**[1]

the date of receipt and acceptance should be inserted later

Abstract In this study we treat scribbling motion as a compositional system in which a limited set of elementary strokes are capable of concatenating amongst themselves in an endless number of combinations, thus producing an unlimited repertoire of complex constructs. We broke the continuous scribblings into small units and then calculated the Markovian transition matrix between the trajectory clusters. The Markov states are grouped in a way that minimizes the loss of mutual information between adjacent strokes. The grouping algorithm is based on a novel markov-state bi-clustering algorithm derived from the Information-Bottleneck principle. This approach hierarchically decomposes scribblings into increasingly finer elements. We illustrate the usefulness of this approach by applying it to human scribbling.

Keywords human movement, movement trajectory, movement primitives, clustering, information bottleneck.

Corresponding author:
Jacob Goldberger
School of Engineering, Bar-Ilan University
Ramat-Gan 52900, Israel
email: goldbej@eng.biu.ac.il
phone: +972-35317053
fax: +972-37384050

(1) The Gonda Brain Research Center, Bar-Ilan University, Ramat-Gan, Israel. Tel.: +972-3-5317755, Fax: +972-3-5317755 · (2) School of Engineering, Bar-Ilan University, Ramat-Gan, Israel

1 Introduction

Human beings and monkeys have an almost endless repertoire of hand movements. When scribbling, for instance, we can generate complicated forms by continuously moving our hand for several seconds. It is generally assumed that we produce these complicated drawings by concatenating a limited set of elementary shapes in a variety of ways, rather than having a distinct brain representation for all the drawings we produce. The current work assumes further that complex drawings are compositional - that is, a very limited set of strokes is used to build a larger set of more complex shapes, which again can be used to build a larger set of more complex shapes and so on. Hand movements would thus be made up of basic movements elements. Utilizing this approach, recording the activity of many neurons in the brain of monkeys while they scribble could shed light on the brain processes associated with such a compositional system. A necessary precondition however is to break down drawings into basic movements. We show that a non-discrete scribbling can be analyzed into such elements, which raises the problem of how to break down non-discrete signals into discrete elements. The basic elements would be simple drawing segments, which would be analyzed as bundles of features, which might well have different physical instantiations in different physical contexts. In this paper we examine how scribbling may be broken down into simple elements and the rules of concatenation involved. Simple movement elements which may be combined to form complex motions are sometimes referred to as 'movement primitives'. For instance, when moving the hand between pairs of targets, human subjects and monkeys tend to generate straight hand paths with single-peaked, bell-shaped velocity profiles. These so-called stereotyped movements are invariant even after changing size, rotation, translation and temporal scaling [8]. These basic movements may be executed sequentially or concurrently (through parameterized superposition), to create a large movement repertoire [15]. If the movement target is shifted during 2D movement, the arm trajectory becomes curved in such a way that there is a vectorial summation of two basic movement elements. This summed trajectory can then be smoothed and expressed as a cost function [6]. Over the years, different motion components have been put forward as candidate primitives. One of the first findings was that it only takes a dozen components to encode a frog's entire motor repertoire [5]. Studies have shown that the frog and the rat's complex limb movements may be generated by a vectorial summation of modular force fields in the spinal cord [2], [17]. Thoroughman and Shadmehr [25] showed that humans learn the dynamics of reaching movements through a flexible combination of primitives that have Gaussian-like tuning functions encoding hand velocity. They argued that activity of cells in the cerebellum may encode primitives that underlie the learning of dynamics. These findings suggest that modular primitives are part of both the planning and the execution of multi-joint limb movements. One year old children's movements can be decomposed into a sequence of stereotyped movements each resembling the simple basic movements of adults [11],[12],[1]. Stroke patients' initial movements can easily be decomposed into simple elementary movements with invariant velocity profiles (duration vs. amplitude) [13]. Sosnik et al. [22] determined whether there is a point in time ("point of no return") in which the generation of a planned action is inevitable. Human subjects were requested unexpectedly to impede free scribbling movement. Their findings suggested that the "point of no return" phenomenon in humans may reflect a high level kinematic plan and could serve as a new operative definition of motion primitives. Recent work done by Polyakov et al. [18] hints that parabolas are used as primitives in monkeys' scribbling. A detailed discussion of the issue of primitives and their composition can be found in the review by Flash and Hochner [7]. Although multiple studies, aimed at unraveling characteristics of primitives have been carried out, very little is known about the rules governing the concatenation of such building blocks into complex movements.

In this paper we examine how scribbling can be broken down into simple elements and describe the rules of concatenation involved. We then apply a novel algorithm to samples of human scribbling. We start by breaking a drawing into small strokes and model each stroke using the directions of ten fractions along the drawing. We consider that primitive movements are actually a bundle of features, and thus treat the set of directions as a set of movement features. Next we apply a Mixture-of-Gaussians (MoG) to group the strokes and compute the Markovian transition probability between groups. Then, using a novel information-theoretic clustering algorithm we group the Markov states in a way that minimizes the loss of mutual information between the current stroke and the upcoming one. We demonstrate the proposed method by applying it to human scribbling.

The article has two parts. In the first part we present and discuss the proposed clustering algorithm. In the second part we present and discuss the application to human scribbling. Section 2 introduces the Markov clustering problem and an efficiently computed algorithm. Section 3 briefly reviews the Mixture-of-Gaussian modeling. Section 4 describes the experimental setup and the pre-processing that was applied to the hand-movement data. Section 5 presents the results obtained by applying the algorithms in Section 2 to the scribbling data. Finally section 6 discusses the findings.

2 The Markov-State Clustering algorithm

2.1 Notation and model

The process of forming primitives of hand movements, by combining groups of similar-strokes into clusters according to their behavior along the time axis, is translated in our approach into the mathematical problem of clustering the states of a Markov process. In this study we propose an efficient information-theoretic clustering algorithm for this clustering task.

Let $X = \{x_t\}$ be an n-valued stationary first order ergodic Markov process defined by the $n \times n$ transition matrix A where $A_{ij} = p(x_1 = j|x_0 = i)$. A function $\pi : \{1, ..., n\} \rightarrow \{1, ..., m\}$, $m \leq n$ defines a partition of the state-space $\{1, ..., n\}$ into m subsets $w = \{w_1, ..., w_m\}$ such that $w_k = \pi^{-1}(k), \quad k = 1, ..., m$. Utilizing π we can define a new m-state lumped Markov process $Y = \{y_t\}$ in the following way. Define $(y_0, y_1) = (\pi(x_0), \pi(x_1))$, i.e. the joint distribution of y_0 and y_1 is defined via the following Markov chain relation:

$$\begin{tikzcd} X_0 \arrow[r, "p(x_1|x_0)"] \arrow[d, "\pi"'] & X_1 \arrow[d, "\pi"] \\ Y_0 & Y_1 \end{tikzcd}$$

More explicitly, the joint distribution of y_0 and y_1 is:

$$p(y_0 = k, y_1 = l) = \sum_{i \in w_k} \sum_{j \in w_l} p(x_0 = i, x_1 = j)$$

Since x is a stationary process, it can be easily verified that the marginal distributions of y_0 and y_1 are the same. Let Y be the stationary m-state Markov process defined by the stochastic matrix $p(y_1|y_0)$.

Our goal is to cluster the states of the process X to form a new reduced Markov process that best preserves the structure of the original process X. Intuitively, states i and j are viewed as similar if both the future conditional distributions $p(x_1|x_0=i)$ and $p(x_1|x_0=j)$ and the past conditional distributions $p(x_0|x_1=i)$ and $p(x_0|x_1 = j)$ are similar. The Information-Bottleneck (IB) principle [26] can be used to formalize this intuition. The IB principle for this case states that the best clustering function π of the n states into m clusters is the one that maximizes the mutual information $I(\pi(x_0);\pi(x_1)) = I(y_0;y_1)$ over all the partitions of the state-space into m subsets. Utilizing a standard information-theory manipulation we can derive several equivalent forms for the cost function we want to minimize.

$$\begin{aligned} C(\pi) &= I(x_0;x_1) - I(y_0;y_1) \qquad (1)\\ &= D(p(x_1|x_0)\|p(x_1|y_0)) + D(p(y_0|x_1)\|p(y_0|y_1))\\ &= D(p(x_0,x_1)\|p(y_0,y_1)p(x_0|y_0)p(x_1|y_1))\\ &= H(y_0,y_1) + H(x_0|y_0) + H(x_1|y_1) - H(x_0,x_1) \end{aligned}$$

where $y_0=\pi(x_0)$, $y_1=\pi(x_1)$, D is the Kullback-Leibler divergence and H is the entropy function. The optimal state-clustering is the one that minimizes the information-loss function $C(\pi)$.

2.2 The Clustering Algorithm

There is no closed-form solution for the optimization problem posed in Section 2. Several standard optimization algorithms can be utilized to find the best clustering. We can use an agglomerative algorithm based on a bottom-up merging procedure. We can use a K-means clustering algorithms on which the Bregman distance is the Kullback-Leibler divergence. Alternatively we can apply a greedy sequential algorithm that can be viewed as a sequential version of the K-means algorithm (see e.g. [19]). In this study we apply the sequential greedy algorithm which has been found to perform well in terms of both clustering quality and computational complexity. The sequential clustering algorithm starts with a random clustering of the states into m clusters. We go over the n original states in a circular manner and check for each state whether its removal from one cluster to another can increase the cost function $I(y_0;y_1)$.

The basic step in each of these algorithms is composed of computing the distance function between two clusters. It can be verified that in our case the information-bottleneck principle implies that this distance is the information-loss caused by merging the two clusters into a single one; i.e. the difference between mutual-information of the reduced Markov processes before and after the two clusters are merged. We then derive an explicit and efficiently computed expression for this distance between clusters.

Assume we are given a partition of the Markov states $w = \{w_1,...,w_m\}$ and we want to compute the information loss caused by merging the clusters w_1 and w_2 to obtain a new reduced partition $w' = \{w_1 \cup w_2, w_3, ..., w_m\}$ into $m-1$ clusters. Let y_0 and y_1 be the Markov chain variables defined by w and let y'_0 and y'_1 be the Markov chain variables defined by w'. The information-loss can be efficiently computed in the following way.

$$\begin{aligned}
&d(w_1, w_2) = I(y_0; y_1) - I(y^{'}_0; y^{'}_1) = && (2)\\
&I(y_0; y_1) - I(y_0; y^{'}_1) + I(y_0; y^{'}_1) - I(y^{'}_0; y^{'}_1) = &&\\
&\sum_{i=1,2} p(w_i) D(p(y_0 | y_1 = w_i) || p(y_0 | y_1 \in w_{12})) + &&\\
&\sum_{i=1,2} p(w_i) D(p(y^{'}_1 | y_0 = w_i) || p(y^{'}_1 | y_0 \in w_{12})) = &&\\
&p(w_{12})(JS(p_{1|0}(\cdot|w_1), p_{1|0}(\cdot|w_2)) + && (3)\\
&\qquad JS(p_{0|1}(\cdot|w_1), p_{0|1}(\cdot|w_2))) - p_{01}(w_{12}, w_{12}) I_{12} &&
\end{aligned}$$

where $w_{12} = \{w_1, w_2\}$ and $p_{01}, p_{1|0}, p_{0|1}$ are the joint and conditional distributions of the reduced-Markov random-variables $y^{'}_0$ and $y^{'}_1$. JS is the Jensen-Shannon divergence. $I_{12} = I(y_0; y_1 | y_0, y_1 \in w_{12})$ is the mutual information of the following joint-distribution matrix:

$$\frac{1}{p_{01}(w_{12}, w_{12})} \begin{pmatrix} p_{01}(w_1, w_1) & p_{01}(w_1, w_2) \\ \\ p_{01}(w_2, w_1) & p_{01}(w_2, w_2) \end{pmatrix} \tag{4}$$

where $p_{01}(w_1, w_2)$ means $p(x_0 \in w_1, x_1 \in w_2)$ etc.

Hence, the distance measure $d(w_1, w_2)$ takes into account both the future and past conditional distributions. The possible overlap between these two distance components is subtracted from the sum. The sequential clustering algorithm requires the computation of the change in the cost function when moving a state from one cluster to another which can be efficiently done using expression (3).

One drawback of the sequential algorithm (in contrast to agglomerative approaches) is that the number of clusters must be given as input to the algorithm. We can slightly modify the algorithm in such a way that we can simply provide a rough estimation (upper bound) on the number of desired clusters. Consider the case of a cluster that contains a single object s. The iterative-sequential algorithm will not merge s into any other cluster because obviously this cannot increase the cost function $I(y_0; y_1)$. The algorithm will always prefer to leave s as a single member of a cluster. In the modified version we enforce a singleton cluster to be merged into another cluster. This step reduces the number of clusters by one. Utilizing this scheme, the number of output clusters can be adapted to the data. Note that in this method each of the output clusters must contain at least two members. The algorithm is summarized in Table 1. Since there is no guarantee that the algorithm will find the global optimum, we can apply the algorithm on several random partitions and choose the best local optimum.

We conclude this section with a short discussion on works related to the clustering algorithm described above. Information-theoretic approaches have been intensively used for clustering and co-clustering methods [26,4,20]. Unlike previous works, in our setup the same clustering function π is simultaneously applied to the two random variables x_0 and x_1. Ge et al. [9] considered the Markov-state clustering problem as a parameter estimation problem of a HMM. They viewed the reduced-state model as a constraint HMM. The original Markov-process is viewed as the observed part of the HMM and the constraint is that each observed symbol can appear only in one hidden-state. The similarity between this approach and ours is related to the analogy between

```
**Input**: A Markov transition matrix $n \times n$.
**Output**: A partition of the Markov states into (at most) $m$ clusters.

Algorithm:
  1. Choose a random partition of the Markov state into $m$ clusters.
  2. Loop until there is no change
       – for $s = 1, ..., n$
           – Remove state $s$ from its current cluster.
           – If $s$ is the only member of its cluster, delete the cluster.
           – Merge $s$ into the cluster $w_k$ that minimizes the distance $d(\{s\}, w_k)$. (see expression (3)).
```

Table 1 The Markov-states clustering algorithm

EM and IB described in [21]. Note that in our derivation (unlike similar approaches [9]) there is no need to recompute the entire score (whose computational complexity is $O(n^2)$) for each sequential update. The distance measure $d(w_1, w_2)$ is only based on a small part of the transition matrix related to the states in $w_1 \cup w_2$ and it can be computed in $O(n)$ operations. Meila and Shi [16] demonstarted the connection between spectral clustering and clustering the states of a Markov transition matrix of a random-walk process defined by the pairwise-distance matrix. In their approach a cluster of states is characterized as follows. Once the process is in one of the members of the cluster it tends to remain in the cluster.

3 The Gaussian Mixture Model

In this study we utilize the Mixture of Gaussians model (MoG) to cluster a set of strokes into groups based on a feature set representation. This is done as a preprocessing step before we apply the clustering algorithm described above. For the sake of completeness we provide a brief review of the MoG model. Generally, the distribution of a d-dimensional random variable is a mixture of k Gaussians if its density function is a weighted sum of Gaussian densities:

$$f(x) = \sum_{j=1}^{k} \alpha_j \frac{1}{\sqrt{(2\pi)^d |\Sigma_j|}} \exp\{-\frac{1}{2}(x-\mu_j)^T \Sigma_j^{-1} (x-\mu_j)\} \qquad (5)$$

where α_j are the probabilities of occurrence of each Gaussian and μ_j, Σ_j, are the mean and the covariance matrix of each Gaussian respectively. In our implementation we further assume that the covariance matrices are scalar and all the components share the same matrix. In this case expression (5) is reduced to

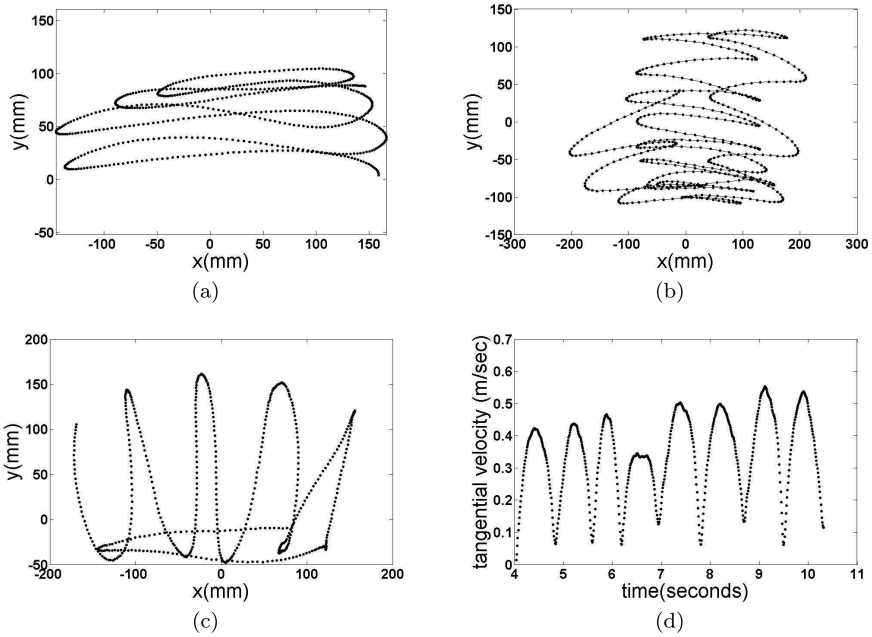
$$f(x) = \frac{1}{(2\pi\sigma^2)^{d/2}} \sum_{j=1}^{k} \alpha_j \exp(-\frac{1}{2\sigma^2}\|x-\mu_j\|^2) \qquad (6)$$

where σ^2 is the variance parameter. Given a set of samples $x_1, ..., x_n \in R^d$ we can utilized the well known iterative EM algorithm [3] to find the maximum likelihood parameter set.

4 Experimental Setup and Data Pre-Processing

Male participants in their 20s were trained for ten blocks, one minute each, spaced two minutes apart. The participants controlled the position of a 5 mm diameter green circle cursor with a horizontal 2-jointed low-friction manipulandum. The cursor and workspace were projected on a horizontal board at chest level in front of the participant so that his hand position was mapped directly onto the cursor position. A sheet of plain white paper was placed on the table and prevented the participant from seeing his hand; it also enabled the projected light to be reflected from the board. During the experiment, the manipulandum's angular position was transformed into X-Y coordinates and was recorded at a rate of 100Hz. In order to remove the high frequency, small jerky movements caused by physiological tremor, the data were smoothed by a Gaussian filter, at an 8 Hz cut off frequency. The participant was asked to shift the manipulandum handle freely over the workspace once hearing a "GO" signal and stop upon hearing a "STOP" signal. During the sessions every time an invisible target on the workspace was hit, a short beep was heard and the participant was rewarded with a small amount of money. At the end of the last session the participant received the accumulated amount of money. The invisible target was a 30 mm diameter circle randomly positioned at one of 23 potential locations and was re-positioned every time it was hit by the participant. If a target was not hit within five seconds, its position changed randomly to a new location. The cursor's position was monitored and traced so that the outcome looked like a scribbled drawing [22]. Figure 1 shows the kinetic behavior of a hand trajectory (scribbling) of three participants (G, O and D) in a 6 second section.

Using the tangential velocity profile (Figure 1d), we segmented the trajectory so that each stroke's initial velocity would correspond to a local minimum or maximum velocity along the entire trajectory. Naturally, each stroke's final velocity corresponds to the next local maximum or minimum velocity along the entire trajectory. Note that the beginning and the end points of the movement along the velocity profile were regarded as a local minimum. We used the local extrema in the tangential velocity as it was found that these are the points at which the movement speed parameters may abruptly change [27]. The outcome of the segmentation procedure is two sets of strokes - accelerating and decelerating. Next we extracted features from each stroke. Motivated by methods that were successfully applied to on-line hand-writing recognition systems [24], we modeled the strokes using the directions of 10 equidistant (either in their time duration or length) fractions along the drawing. We obtained 10 angles (made by the tangents and the horizon) as a reproducible description of the stroke. Note that the angle parameter is invariant to both size and translation of the input stroke. Thus, each stroke was represented as a single point in a 10-dimensional feature space. Our goal was to find the minimal number of fractions and angles that would enable us to reconstruct the original strokes in a reliable manner. For linear strokes, one fraction would enable us to successfully guess the original one. On the other hand, curved strokes demand a higher number of fractions and the original strokes can never be depicted precisely. Note that all the strokes should share the same number of fractions. Figure 2 depicts the mean square error between the original strokes and the reconstructed ones given different numbers of fractions. We concluded that 10 angles sufficed to minimize the stroke reconstruction error to a satisfatory level. It should also be noted that although figure 2 presents only participant G's reconstruction error, this behavior



(a)

(b)

(c)

(d)

Fig. 1 A scribbling example 6 seconds of hand movement (scribbling) of three participants: (a) Participant G, (b) Participant O, and (c) Participant D. (d) 6 seconds of the tangential velocity as a function of time, using the scribbling of participant G. The other two participants exhibited similar behavior.

was common to all the participants. From now on we apply our mathematical analysis to the reconstructed strokes.

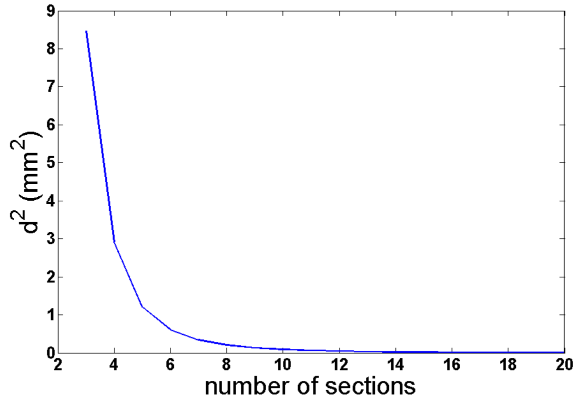
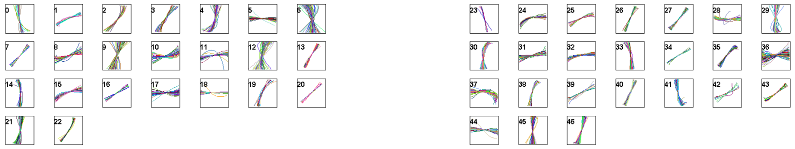


Fig. 2 The reconstruction error is the sum of the squared differences between the original and the reconstructed strokes of participant G.. The reconstruction error monotonically decreases as the number of fractions increases and saturates at 10 fractions. More than 10 fractions do not bring the reconstructed strokes much closer to the original ones. Note that the same results where obtained for all the other participants.



(a)

(b)

Fig. 3 A plot of all the accelerating and the decelerating groups of participant G. Each subplot describes the reconstructed strokes (not the original ones) from the 10 angles, thus giving all the strokes the same size and position. (a) Accelerating states. (b) Decelerating states. The hand movement strokes in each group (plotted in a box) were aligned according to the average (center-of-mass) point.

5 Experimental Results

We clustered each of the two sets of strokes (accelerating and decelerating) into groups based on the angle-vectors representation. Each stroke was represented by a 10-dimensional vector. To simplify the model and to avoid over-fitting, we assumed that every one of the 10 angles in each stroke was sampled independently from the rest, even though it is clear that smooth hand movement relies on a dependency among adjacent angles. Thus, we restricted the covariance matrices to be diagonal. We further constrained all the Gaussian variances across all dimensions and across all Gaussian components to be the same. The MoG model, therefore, consists of scalar covariance matrices which are all the same. This results in a single variance parameter that needs to be learned from the data.

In order to find the maximum-likelihood parameter-set of the MoG model, we used the EM algorithm. Additionally, the M-step is affected by the constraints that we imposed on the covariance matrices. We put the feature vectors into groups. The number of groups was empirically chosen as the minimal number of groups that could model the variability in the stroke data. The variance parameter that was learned throughout by the EM algorithm was $\sigma^2 = 0.01$. In other words, the standard deviation was found to be 0.1 radians. To assign the feature vectors into groups we had to consider the cyclic nature of angles. Hence, angles such as 1 degree and 359 degrees were considered to be close to one another. While computing the Gaussian density of a stroke we always computed the distance between the mean angle and the observed angle along the direction that yielded a smaller distance. Since the variances were found to be much smaller than 360 degrees, this makes sense. After the MoG was learned we could label the strokes. We assigned each stroke to the Gaussian component with the maximal posterior-probability. The groups' labels were used to estimate a Markov transition matrix. Figure 3 presents a set of states obtained from the EM algorithm. Every subplot depicts a state derived from the MoG model. The strokes are not all alike though most of them have the same structure as can be expected when dealing with a Gaussian distribution. Figure 3 presents participant G's reconstructed strokes.

The Markovian transition matrix presented in Figure 4 is based on participant G's results. It describes the probability that the next stroke will belong to state i, given that the present stroke belongs to state j. The Markovian matrix reveals a 4 quadrant probability structure, where the non-zero probabilities are contained within the two off-diagonal quadrants. This special structure can be accounted for by the fact that states 0-22 are states of accelerating strokes whereas states 23-46 are states of decelerating strokes, and strokes from one group must follow and be followed by the strokes from the other.

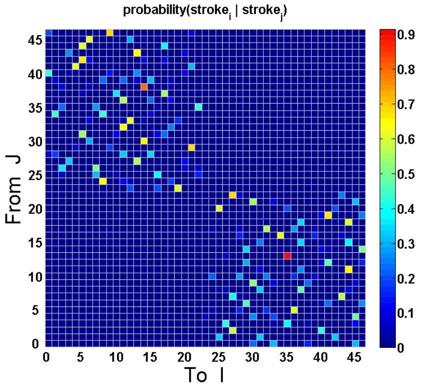


Fig. 4 The Markov transition matrix of participant G. Every cell in the transition matrix represents the conditional transition probability between two states. The color of each cell is equivalent to its two-states transition probability. For example, the red cell in (a) indicates that for a given stroke from the 13th state, the probability of the next one being taken from the 35th state is about 90%. The probabilities for each row sum up to 1. The two-block structure corresponds to the notion that accelerating strokes should precede decelerating strokes and vice versa.

Our next step was to group the states of the Markovian process into clusters. The clustering algorithm described in Section 3 was utilized for this task. This algorithm was implemented on those strokes that were assigned to a specific state based on an equal time division or based on an equal length duration. Figure 5 presents the mutual-information (MI) clustering-quality score as a function of the number of clusters for the three participants. As can be seen, we obtained a flattened curve, i.e., in the range of 8-12 clusters(participant G) or 6-10 clusters (participants D and O) a significant decrease in the MI took place. Twelve (out of 8-12) and 8 (out of 6-10) clusters were selected as they seem to generate the clearest structure in the grouped transition matrices.

When reordering the rows and columns of the matrix in Figure 4 so as to have all states of strokes within one cluster occupy adjacent rows (columns) we obtained Figure 6. Figure 6 presents a block structure Markovian transition matrices. The block structure corresponds to the notion that states that share the same clusters are adjacently positioned. Following the transitions with the highest probabilities, shown in Figure 6a, it is clear that a stroke from cluster 1 is most likely to be followed by a stroke from cluster 2, which in turn is most likely to be followed by a stroke from cluster 3, which is most likely to be followed by a stroke from cluster 4. The next most likely stroke to follow belongs to cluster 1 again; thus clusters 1, 2, 3, and 4 form a cycle which

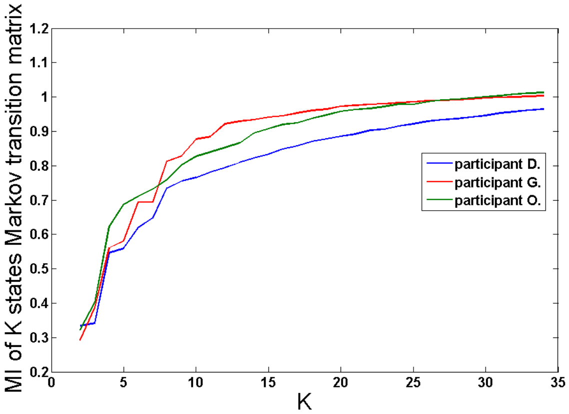


Fig. 5 The mutual-information (MI) score of the clustered Markov process as a function of the number of clusters for the three participants (participants G, O and D).

tends to repeat itself. Similarly, clusters 5, 6, 7, 8 and 9, 10, 11, 12 form two additional cycles.

Sequences of clusters, like patterns of sounds that tend to follow each other with high probability, are like words, and the tendency of each such sequence to repeat several times is like a phrase. The matrices in Figure 6 also indicates the structure of transition between phrases. For example, in Figure 6a, the transition from a phrase constructed from repetitions of word 1 (clusters 1, 2, 3, 4) into a phrase constructed from repetitions of word 2 (clusters 5, 6, 7, 8) occurred almost solely from the first state of cluster 3 to the second state of cluster 8 (the small cyan square in row 7 and column 26).

Figure 7 presents a set of participant G's twelve clusters, in a twelve row structure, obtained as a result of the clustering algorithm. Each cluster has a geometry structure, specific direction and acceleration orientation.

6 Discussion and Conclusion

The results presented in Section 5 illustrate that the algorithm described here can successfully reveal the inner structure of the scribbling data.

However, several somewhat arbitrary decisions were made. Initial parsing of motion into very small strokes may be unjustified. A simple intuition suggests that each element of motion has a bell-shaped velocity profile and therefore, breaking down motion between minima of tangential velocity might be more adequate. However, psychophysics indicates that different motion elements may be concatenated at inflection points where velocity is maximal [27]. The recent discovery of parabolas as putative primitives also shows concatenation at points of maximal velocity but without inflection in curvature ([18]). Thus breaking at both minima and maxima of tangential velocity are more appropriate. In cases where such a break constitutes only one half of an elementary motion, this would show up easily in the Markovian transition matrix (e.g. the red cell in Figure 4a). Thus, there is no claim that the small strokes presented in Figures 3 and 7 are 'primitives' of any sort.

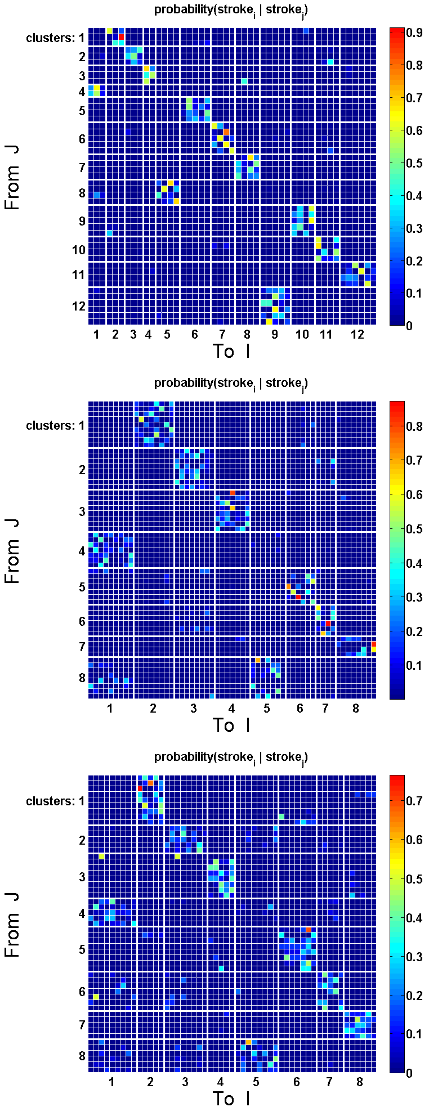


Fig. 6 Markov transition matrices. The states from the matrix in figure 4 were rearranged so that states that are members of the same cluster correspond to adjacent rows (and columns). (a) participant G.: The 12 block structure corresponds to the 12 clusters of states forming the outcome of the clustering procedure. Clusters $< 1, 2, 3, 4 >$, $< 5, 6, 7, 8 >$, $< 9, 10, 11, 12 >$ form 3 cycles that could be viewed as words in a language. (b) and (c) participant O and D respectively: 8 block structure of clusters $< 1, 2, 3, 4 >$, $< 5, 6, 7, 8 >$ form 2 cycles.

Taking angles as the features which describe each small stroke is arbitrary as well. However, in view of the notion that activity of neurons in the upper-arm areas of the motor cortices are related mostly to the direction of arm motion [10], [23], this choice seems reasonable. Neglecting the position and size of the motion is somewhat limiting

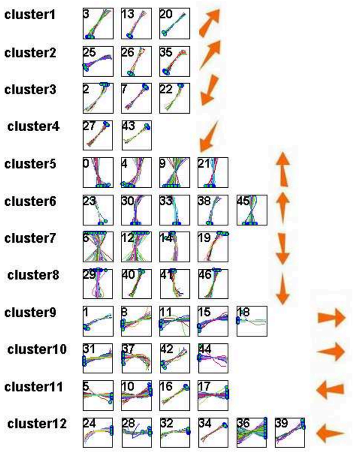


Fig. 7 Participant G's 12 clusters of Markov states, based on equal-distance division. Each row represents a cluster whereas each subplot represents a state. For the sake of simplicity the orange arrows indicate the movement and acceleration orientation.

as will be discussed later. Selecting 10 points along each stroke seems justified as seen from the low approximation error achieved with 10 points (Figure 2). When the points were taken at equal distances along the stroke, the error was somewhat larger compared to 10 equi-time points. This can be understood in view of the psychophysics of motion where speed is lower for higher curvature [14]. Thus, in equi-time sampling there are more dense points at regions of high curvature and the fidelity of stroke reconstruction is higher. The initial parsing from low to high speed and from high to low speed dictates treating the accelerating and decelerating strokes separately.

Treating the vectors of 10 angles as a mixture of Gaussians with no correlations between angles is definitely inaccurate. The strokes are very smooth, which means there is a high correlation between adjacent angles. We used this treatment because it greatly simplified the classification and the end result (Figure 3 showed a reasonably good separation into classes). There are also a few aberrant strokes (e.g. the highly curved strokes in classes 11 or 28 in 3). These are the outcome of forcing every stroke to belong to some class. It might have been wiser to exclude such exceptional strokes from the study. However as there were only very few of them their inclusion did not affect the results. Using the proposed method for clustering the Markov states seemed to work well.

However, the transition from shallow to steep decline in mutual information with a decreasing number of clusters (Figure 5) is gradual in the range of 8-12 clusters(participant G) or 6-10 clusters (participants D and O). Considering the way we parsed the motion into strokes, it is reasonable to assume that the clusters should include an even number of states (sets of accelerating-decelerating pairs). Selecting 12 clusters for participant G and 8 clusters for participants D and O seemed to generate the clearest structure in the grouped transition matrices in Figure 6.

Careful examination of these matrices (Figure 6) reveals a great deal about the internal structure of the scribbling. As noted in Section 4, all the participants' clusters could be organized into a words structure. Participant G's clusters could be organized into 3 words, composed of clusters 1, 2, 3, 4, 5, 6, 7, 8, and 9, 10, 11, 12. Let us refer to them as words w_1, w_2, and w_3. The matrix reveals that each of these words tends to repeat itself, such that there are 'phrases' composed of $< w_1, w_1, w_1, >$, $< w_2, w_2, w_2, >$, and $< w_3, w_3, w_3, >$. Let us call them p_1, p_2, and p_3.

Participants O and D scribblings were sorted into 2 words, w_1 composed of 1,2,3,4 and w_2 composed of 5,6,7,8 which again could be organized in a repeating alternate phrases p_1 and p_2.

What remains unclear is which of the four classes in each word begins the phrase cycle. The transitions between participant G's three phrases form a paragraph structure. p_1 is a transition phrase from which the drawing may continued either by p_2 or p_3. Both p_2 and p_3 terminate by going back to p_1, but there are no direct transitions between p_2 and p_3. Thus these transitions between words and phrases can be regarded as the syntax of the drawing language. Note that the transitions into and out of phrases are confined to individual allophones (single cell in the matrix). The general structure of the drawing emerges as follows. p_2 is up and down strokes. It ends by an oblique lower-left to upper-right strokes and back (p_1). p_3 is left and right strokes and it too ends by the oblique p_1 phrase. This can actually be seen by observing 25 seconds of scribbling.

However, participant G also scanned the entire workspace by gradually moving from one side to the other, and only shifted from one scanning style (phrase) to another when reaching the edge. The drift and position of the changing style rule did not emerge in our analysis probably because the position of the motion in the workspace was not one of the features used to classify the strokes.

In conclusion, our algorithm can reveal the inner structure of scribbling and the laws of its composition.

In future studies we intend to apply this analysis to experiments in which monkeys scribble while their motor cortex activity is recorded by multiple micro-electrodes. We plan to decompose these recordings using the HMM algorithm to articulate a set of neural states. The idea is to find a match between a set of movement elements and a set of brain activity recording fragments.

Acknowledgements

This work was supported in part by the Deutsch-Israelische Projectkooperation (DIP).

References

1. N. E. Berthier. Learning to reach: a mathematical model. *Dev. Psychol.*, 1996.
2. E. Bizzi, M. C. Tresch, P. Saltiel, and A. dAvella. New perspectives on spinal motor systems. *Nat. Rev. Neurosci.*, 2000.
3. A. Dempster, N. Laird, and D. Rubin D. Maximum likelihood estimation from incomplete data via the em algorithm. *J. Roy. Statist. Soc. Ser. B*, pages 1–38, 1977.
4. I. S. Dhillon, S. Mallela, and D. S. Modha. Information-theoretic co-clustering. *International Conference on Knowledge Discovery and Data Mining(KDD)*, 2003.
5. S. Giszter E. Bizzi and and F. A. Mussa-Ivaldi. Computations underlying the execution of movement: a novel biological perspective. *Science*, 1991.
6. T. Flash and E. A. Henis. Arm trajectory modification during reaching towards visual targets. *J. Cogn. Neurosci.*, 1991.
7. T. Flash and B. Hochner. Primitives in vertebrates and invertebrates. *Curr. Opin. Neurobiol.*, 2005.

8. T. Flash and N. Hogan. Coordination of arm movements: An experimentally confirmed mathematical model. *J. of Neuroscience*, 1985.
9. X. Ge, S. Parise, and P. Smyth. Clustering markov states into equivalence classes using svd and heuristic search algorithms. *AISTATS*, 2003.
10. A. P. Georgopolulos, J. F. Kalaska, R. Caminiti, and J. T. Massey. On the relations between the direction of two-dimensional arm movements and cell discharge in primate motor cortex. *J. Neurosci.*, pages 1527–1537, 1982.
11. C. Hofsten. Structuring of early reaching movements: a longitudinal study. *J. Mot. Behav.*, 1991.
12. J. Konczak, M. Borutta, H. Topka, and J. Dichgans. The development of goal-directed reaching in infants: hand trajectory formation and joint force control. *Exp. Brain Res.*, 1995.
13. H. I. Krebs, M. L. Aisen, B. T. Volpe, and N. Hogan. Quantization of continuous arm movements in humans with brain injury. *Proc. Natl. Acad. Sci.*, 1999.
14. F. Lacquaniti, C. Terzuolo, and P. Viviani. The law relating kinematic and figural aspects of drawing movements. *ACTA Psychol*, pages 115–130, 1983.
15. M. J. Mataric. Sensory-motor primitives as a basis for imitation: Linking perception to action and biology to robotics. imitation in animals and artifacts. *The MIT Press*, 2001.
16. M. Meila and J. Shi. A random walks view of spectral segmentation. *AISTATS*, 2001.
17. F. A. Mussa-Ivaldi and E. Bizzi. Motor learning through the combination of primitives. *Philos Trans R Soc Lond B Biol Sci*, 2000.
18. F. Polyakov, R. Drori, M. Abeles, and T. Flash. Parabolic primitives and dimensionality reduction through practice: analysis and mathematical modeling of monkey scribbling movements. submitted.
19. N. Slonim, N. Friedman, and N. Tishby. Unsupervised document classification using sequential information maximization. *ACM SIGIR*, pages 129–136, 2002.
20. N. Slonim, N. Friedman, and N. Tishby. Multivariate information bottleneck. *Neural Computation*, pages 1739–1789, 2006.
21. N. Slonim and Y. Weiss. Maximum likelihood and the information bottleneck. *Proc. of Neural Information Processing Systems*, 2003.
22. R. Sosnik, M. Shemesh, and M. Abeles. The point of no return in planar hand movements: an indication of the existence of high level motion primitives. *Cognitive Neurodynamics*, pages 341–358, 2007.
23. E. Stark, R. Drori, I. Asher, Y. Ben-Shaul, and M. Abeles. Distinct movement parameters are represented by different neurons in the motor cortex. *Eur J. Neurosci*, pages 1055–66, 2007.
24. C. C. Tappert. Cursive script recognition by elastic matching. *IBM J. Research and Development, vol. 26*, 1982.
25. K.A. Thoroughman and R. Shadmehr. *Nature*, pages 742–747, Learning of action through adaptive combination of motor primitives.
26. N. Tishby, F. Pereira, and W. Bialek. The information bottleneck method. In *Proc. of the Annual Allerton Conference on Communication, Control and Computing*, 1999.
27. P. Viviani and R. Schneider. A developmental study of the relationship between geometry and kinematics in drawing movements. *Journal of Experimental Psychology*, 1991.