# Segmentation of Human Motion into Dynamics Based Primitives with Application to Drawing Tasks

D. Del Vecchio, R.M. Murray, P. Perona
Division of Engineering and Applied Science
California Institute of Technology
Pasadena, CA 91125

**Abstract**

Using tools from dynamical systems and systems identification we develop a framework for the study of decomposition of human motion. The objective is understanding human motion by decomposing it into a sequence of elementary building blocks, which we refer to as *movemes*, which belong to a known alphabet of dynamical systems. We develop classification and segmentation algorithms with error analysis and we test them on human drawing data.

## 1 Introduction

Building systems that can detect and recognize human actions and activities is an important goal of modern engineering. Applications range from human-machine interfaces to security to entertainment. A fundamental problem in detecting and recognizing human action is one of representation. Our point of view is that human activity should be decomposed into building blocks which belong to an "alphabet" of elementary actions; for example the activity "answering the phone" could be decomposed into the sequence "step-step-step-reach-lift", where "step", "reach" and "lift" may not be further decomposed. We refer to these primitives of motion as *movemes*. Our aim is then to build an alphabet of movemes which one can compose to represent and describe human motion similar to the way phonemes are used in speech. The word "moveme" intended as primitive of motion was invented by [2]. They studied periodic or stereotypical motions such as walking or running where the motion is always the same and therefore their movemes, like the phonemes, were repeatable segments of trajectory. [7] studied motions that were parametrized by an initial condition and a target. They proposed that movemes ought to be parametrized by goal and style parameters. Their moveme models are phenomenological and non-causal.

In [4] the authors defined movemes in terms of causal dynamical systems and developed an elementary off-line segmentation algorithm for 2D motion, based on an alphabet of two movemes, and activities composed at most of two movemes. In this paper we attempt to develop off-line classification and segmentation algorithms, which, based on an alphabet of movemes, find the sequence of switching times between movemes and their class in a trajectory composed by a sequence of an unspecified number of movemes. The dynamical systems representation for describing human motion and the segmentation idea are not novel, some sample citations include [13, 11, 12]. Our contribution lies mainly in the detailed treatment of the error of the algorithms proposed, which is possible thanks to the dynamical systems framework used.

In Section 2 we recall some basic definitions and introduce the classification problem. In Section 3 and 4 we set up the segmentation problem. The problem of segmenting data streams originating from different unknown or partially known processes which alternate in time is a general problem of interest to various areas, see for example [8, 9, 6]. We propose a solution to the problem in our particular scenario in which each one of the segments has been generated from the perturbed version of a linear dynamical system belonging to a finite known set of possible linear models. By using system identification techniques [10, 14] and pattern recognition techniques [1, 15] we develop an off-line segmentation and classification algorithm and provide an analytical error analysis. In Section 5 we show the results of the algorithm on the segmentation and classification problem of human drawing data.

## 2 Dynamical Definition of Moveme

We recall in this section a relaxed version of the definition of moveme already presented in [4], we introduce the model class, and we set the classification problem.

### 2.1 Definitions and properties

Let $M(\Theta)$ denote a linear time invariant (LTI) system class parameterized by $\Theta \in E$, $E$ a linear space, and let $\mathcal{U}$ denote a class of inputs. Let $y(t) = Y(M(\Theta)|_{u,x_0})(t)$, for $t \geq t_0$, denote the output of $M(\Theta)$ once parameter

$\Theta \in E$, input $u \in \mathcal{U}$, and initial conditions $x_0$ have been chosen. Let $\theta \in E' \subset E$ be a parameter lying in a subspace of $E$, and define a map $\Upsilon : E \to E'$. We write $\theta = \Upsilon(\Theta)$ to represent the transformation from $\Theta \in E$ to the reduced set of parameters $\theta \in E'$.

**Definition 2.1** Let $M^1 = \{M(\Theta)|\theta \in \mathcal{C}^1\}$ and $M^2 = \{M(\Theta)|\theta \in \mathcal{C}^2\}$ denote two subsets in $M$ with $\mathcal{C}^j \subset E'$ for $j = 1, 2$. $M^1$ and $M^2$ are said to be *dynamically independent* if

(i) the class of systems $M$ and the class of inputs $\mathcal{U}$ are such that

$$Y(M(\Theta_1)|_{u_1,x_0})(t) = Y(M(\Theta_2)|_{u_2,x_0})(t), \quad \forall t \geq t_0$$

if and only if $(\Theta_1, u_1) = (\Theta_2, u_2)$ for $u_1 \in \mathcal{U}$ and $u_2 \in \mathcal{U}$;

(ii) the sets $\mathcal{C}^1$ and $\mathcal{C}^2$ are non empty, bounded, and have trivial intersection, i.e. $\mathcal{C}^1 \bigcap \mathcal{C}^2 = \{\emptyset\}$.

Each of the elements of a set $\mathcal{M}$ of mutually dynamically independent model sets is called a *moveme*.
In this paper, we choose our model class $M$ and input $u$ as asymptotically stable linear systems driven by a unit step input with full state output:

$$\begin{aligned} \dot{x} &= Ax + b \\ y &= x \ , \end{aligned} \tag{1}$$

where $A \in \mathbb{R}^{n\times n}$, $x = (x_1, ..., x_n) \in \mathbb{R}^n$, $b \in \mathbb{R}^n$, so that $\Theta = (A|b) \in E = \mathbb{R}^{n\times(n+1)}$ and $\theta = A \in E' = \mathbb{R}^{n\times n}$, with $\Upsilon(A|b) = A$.

**Assumption 2.1** Given $x(t)$ as the output of model (1) we assume that the initial condition $x_0$ is such that for any $v \in \mathbb{R}^{n+1}$, $v^T\overline{x}(t) = 0$, $t \in [t_1, t_2]$, $t_2 > t_1 \implies v = 0$, where $\overline{x} = (x^T, 1)^T$.

According to this assumption there is a one-one correspondence between $x(t)$ and parameters $(A|b)$ of model (1), so that we have the following lemma whose proof can be found in [3].

**Lemma 2.1** *Let $x(t)$ and $z(t)$ be generated by two LTI systems*

$$\dot{x} = A_1x + b_1 \qquad \dot{z} = A_2z + b_2 \tag{2}$$

*and let Assumption 2.1 hold. Then $z(t) = x(t)$ for all $t$ if and only if $(A_1|b_1) = (A_2|b_2)$.*

Thus, by this lemma, property (i) of Definition 2.1 is satisfied by our choice of $M$ and $\mathcal{U}$. Property (ii) is verified if we choose for example $\mathcal{C}^j$, $j = 1, ..., m$ as balls in $\mathbb{R}^{n\times n}$ with centers $A_c^j \in \mathbb{R}^{n\times n}$, $j = 1, ..., m$, and radii $r_j$, such that:

$$\begin{aligned} &\mathcal{C}^j = B_{r_j}(A_c^j) \ , && j = 1, ..., m \\ &\mathcal{C}^j \bigcap \mathcal{C}^k = \{\emptyset\}, && j \neq k \end{aligned} \tag{3}$$

where $m$ is the number of movemes and the matrix norm is the Frobenius norm. Given any signal $x(t)$ we can determine a good representative of such a signal in the class of models (1) by minimizing the cost function (see for example [10]):

$$(\hat{A}|\hat{b}) = \arg\min_{(A|b)} \frac{1}{2} \int_{t_0}^{T} (\dot{x} - (A|b)\overline{x})^T(\dot{x} - (A|b)\overline{x})dt \tag{4}$$

with $\overline{x} = (x^T, 1)^T$, so to get the estimate of $x$ in model class (1) as $\dot{\hat{x}} = \hat{A}\hat{x} + \hat{b}$ with $\hat{x}(t_0) = x(t_0)$. In the case in which $x(t)$ has been generated by (1), by virtue of Assumption 2.1 it is easy to check that (4) leads to $(\hat{A}|\hat{b}) = (A|b)$, so that if $A \in \mathcal{C}^j$, for some $j \in \{1, ..., m\}$ we can classify $x(t)$ as output of moveme $M^j$ just by finding $k \in \{1, .., j, ..m\}$ such that $\hat{A} \in \mathcal{C}^k$. The solution is unique because of trivial intersection between sets as specified in (3). The following section addresses the same classification problem when $x(t)$ has been generated by a perturbed version of system (1).

### 2.2 Classification Problem

Let the signal $x(t)$ be generated by

$$\begin{aligned} \dot{x} &= (A_c^j + \delta U)x + b + d(t) \\ y &= x \ . \end{aligned} \tag{5}$$

with $A = A_c^j + \delta U$ with $U$ a unit norm matrix and $A_c^j$ center of $\mathcal{C}^j$, for some $j \in \{1, ..., m\}$ and $d(t)$ is a bounded realization of white noise. Under what conditions on $A$ and $d(t)$ can we still classify $x(t)$ as output of moveme $M^j$? The answer is provided by the following lemma whose proof is in [3].

**Lemma 2.2** *Let $x(t)$, $t \in [t_0, T]$ be generated by (5), where $A_c^j$ is the center of $\mathcal{C}^j$ for some $j \in \{1, ..., m\}$ as in (3). Let $\hat{A}$ be the least squares estimate according to (4). There exist positive constants $\overline{d}$ and $\overline{\delta}$ such that if $\delta \leq \overline{\delta}$ and $\|d(t)\| \leq \overline{d}$, then*

$$arg_{k\in\{1,...j,...m\}}\{\|\hat{A} - A_c^k\| \leq r_k\} = j.$$

In this section we have recalled the definition of moveme and introduced the classification problem. In the next section we use these notions to develop the segmentation algorithm.

## 3 Problem Statement

Consider the sequence of systems for $i = 0, ..., l$

$$\begin{cases} \dot{x} = (A_i + \delta U_i)x + b_i + d(t) & t \in [\tau_{i-1}, \tau_i) \\ \dot{x} = (A_{i+1} + \delta U_{i+1})x + b_{i+1} + d(t) & t \in (\tau_i, \tau_{i+1}] \end{cases} \tag{6}$$

with $x \in \mathbb{R}^n$, $A_i \in \mathbb{R}^{n\times n}$ an unknown matrix whose value can take place in the set of known Hurwitz matrices $\{A_c^1, ..., A_c^m\}$, which are centers of the sets defined

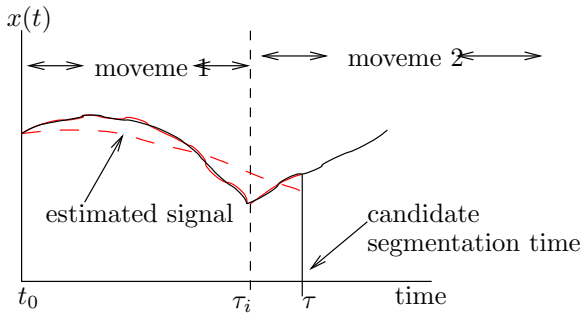


**Figure 1:** Signal considered for computation of approximation and parametric errors (one component shown) and estimated signal $\hat{x}$ (dashed line).

in (3), i.e. $\mathcal{C}^j = B_{r_j}(A_c^j)$ with $\mathcal{C}^j \bigcap \mathcal{C}^k = \{\emptyset\}$ for $j \neq k$, $b_i \in \mathbb{R}^n$ unknown constant vectors, $U_i \in \mathbb{R}^{n\times n}$ norm one matrices (according to Frobenius norm), $\delta \in \mathbb{R}$ modeling uncertainty with $|\delta| \leq \overline{\delta}$, $d(t)$ realization of white noise such that $\|d(t)\| \leq \overline{d}$, $\tau_i$ unknown switching times with $\tau_0$ known starting time and $\tau_l$ known ending time. Consider also the related nominal system:

$$\begin{cases} \dot{x} = A_i x + b_i & t \in [\tau_{i-1}, \tau_i) \\ \dot{x} = A_{i+1} x + b_{i+1} & t \in (\tau_i, \tau_{i+1}] \end{cases} \tag{7}$$

with interconnection condition

$$\frac{\dot{x}(\tau_i^-)^T \dot{x}(\tau_i^+)}{\|\dot{x}(\tau_i^-)\|\|\dot{x}(\tau_i^+)\|} \leq \rho_0 < 1 \ . \tag{8}$$

where we define

$$\dot{x}(\tau_i^-) = \lim_{\tau \to \tau_i^-} \dot{x}(\tau), \qquad \dot{x}(\tau_i^+) = \lim_{\tau \to \tau_i^+} \dot{x}(\tau).$$

The interconnection condition gives a bound on the discontinuity in the trajectory's derivative at the switching points. We wish to obtain sufficient conditions on noise level and parameter uncertainty that allow off-line determination of the sequence of times $\{\tau_1, ..., \tau_{l-1}\}$ and the sequence of matrices $\{A_1, ..., A_l\}$ from the observation of state $x$. If we have a good guess of the switching times, then we can apply Lemma 2.2 so to solve the classification problem in each interval between two switching times. We thus focus our attention on the segmentation part of the problem. We use an iterative approach in which at each iteration we look for the maximizer of a function defined on $[t_0, t_M]$ where $t_M = \tau_l$ and $t_0$ is a starting time which coincides with $\tau_0$ at the first iteration. We want to show the maximizer of such function falls in an interval $I$ around the first switching time encountered after $t_0$; moreover this interval should shrink down to the switching point when noise and parameter uncertainty go to zero. To define such a function we define three quantities for system (6): the approximation error, the parametric error, and the transition factor at time $\tau$. The least squares estimate for $x(t)$, $t \in [t_0, \tau]$, from (4) is $(\hat{A}|\hat{b})(\tau, t_0) = \left[\int_{t_0}^{\tau} \dot{x}\, \overline{x}^T dt\right] \left[\int_{t_0}^{\tau} \overline{x}\, \overline{x}^T dt\right]^{-1}$, which generates the system $\dot{\hat{x}} = \hat{A}(\tau, t_0)\hat{x} + \hat{b}(\tau, t_0)$, with $\hat{x}(t_0) = x(t_0)$. This situation is depicted in Figure 1, where we report the candidate segmentation time $\tau$, the switching time $\tau_i$, the portion of signal under study composed by the sequence of two movemes (solid line between $t_0$ and $\tau$), the estimated trajectory (dashed line). We define the parametric error at time $\tau$ as

$$e_p(\tau, t_0) = \min_{j=1,...,m} \|\hat{A}(\tau, t_0) - A_c^j\|, \tag{9}$$

the approximation error at time $\tau$ as

$$e_a(\tau, t_0) = \frac{1}{\tau - t_0} \int_{t_0}^{\tau} (x - \hat{x})^T (x - \hat{x}) dt \ , \tag{10}$$

and the transition factor is defined as

$$\mathrm{Tr}(\tau) = \tfrac{1}{2}\left(1 - \frac{\dot{x}_{av}(\tau^-)^T \dot{x}_{av}(\tau^+)}{\|\dot{x}_{av}(\tau^-)\|\|\dot{x}_{av}(\tau^+)\|}\right) \tag{11}$$

with $\dot{x}_{av}(\tau^-) := \frac{1}{\Delta\tau}\int_{\tau-\Delta\tau}^{\tau} \dot{x}(t)dt$ and $\dot{x}_{av}(\tau^+) := \frac{1}{\Delta\tau}\int_{\tau}^{\tau+\Delta\tau} \dot{x}(t)dt$, where $\Delta\tau$ is a positive constant depending on perturbation level that will be determined later. Our choice for the function to be maximized is

$$W(\tau, t_0) = \frac{\exp(\frac{-e_p(\tau,t_0)^2}{\sigma^2})\mathrm{Tr}(\tau)}{a + e_a(\tau, t_0)}, \quad \tau \in (t_0, t_M] \tag{12}$$

where $a$ is an arbitrarily small positive constant to prevent the denominator from being zero. By maximizing function $W(\tau)$ we look for the value of $\tau$ that has small approximation error, small parametric error, and a high transition factor. Expression (11) involves integration over time $\Delta\tau$ to attenuate the effect of noise and its expression for system (7) is obtained by letting $\Delta\tau \to 0$. In such a case we find that $\mathrm{Tr}(\tau_i) \geq (1-\rho_0)/2$ and for $\tau \neq \tau_i$, $\mathrm{Tr}(\tau) = 0$. The idea of the transition factor term is to preserve this property as much as possible in the perturbed case so that all the times $\tau_i + \Delta\tau \leq \tau \geq \tau_{i+1} - \Delta\tau$ and $t_0 < \tau \leq \tau_i - \Delta\tau$ are penalized with respect to time $\tau_i$. We also choose to minimize $e_p(\tau, t_0)$ so to reduce the effect of perturbation on the parameter estimates. Alternatively, one could constrain the estimates $\hat{A}$ to lie in a ball around $A_i$, but we do not know the value of $A_i$ *a priori*, we just know that it belongs to a set of possible values. Therefore we decide to minimize the distance of $\hat{A}$ from the closest point $A_j$ at time $\tau$ according to a Gaussian metric.

## 4 Main Result

Consider the sequence of dynamical systems, for $t \in [\tau_0, \tau_l]$, switching at unknown times $\{\tau_1, ..., \tau_{l-1}\}$ defined in (6). We make a number of assumptions on the nominal system (7) and its perturbation:

**Assumption 4.1** The $i^{th}$ segment is of class $j$, with $j$ unknown. In formulas we have $(A_i + \delta U_i) \in \mathcal{C}^j$, and $A_i = A_c^j \in \{A_c^1, ..., A_c^m\}$ for some $j$ and the set of known Hurwitz matrices, $\{A_c^1, ..., A_c^m\}$, is such that $\mathcal{C}^j = B_{r_j}(A_c^j)$ with $\mathcal{C}^j \bigcap \mathcal{C}^k = \{\emptyset\}$ for $j \neq k$.

**Assumption 4.2** $\delta \in \mathbb{R}$ represents modeling uncertainty with $|\delta| \leq \overline{\delta}$, and $d(t)$ is realization of white noise such that $\|d(t)\| \leq \overline{d}$.

**Assumption 4.3** The nominal system

$$\begin{cases} \dot{x} = A_i x + b_i & t \in [\tau_{i-1}, \tau_i) \\ \dot{x} = A_{i+1} x + b_{i+1} & t \in (\tau_i, \tau_{i+1}] \end{cases}$$

satisfies the *interconnection condition*

$$\frac{\dot{x}(\tau_i^-)^T \dot{x}(\tau_i^+)}{\|\dot{x}(\tau_i^-)\| \|\dot{x}(\tau_i^+)\|} \leq \rho_0 < 1 \ .$$

**Assumption 4.4** The state $x(t)$ of the nominal system is such that $v^T \overline{x}(t) = 0, \ t \in [t_1, t_2], \ t_2 > t_1 \quad \Longrightarrow v = 0$, where $\overline{x} = (x^T, 1)^T$.

**Theorem 4.1** *Consider the sequence of dynamical systems given in (6) subject to Assumptions 4.1 to 4.4. Let the function $W(\tau, t_0)$ be defined as*

$$W(\tau, t_0) = \frac{\exp(\frac{-e_p(\tau, t_0)^2}{\sigma^2}) \mathrm{Tr}(\tau)}{a + e_a(\tau, t_0)}, \qquad \tau \in (t_0, t_M]$$

*for $t_0 = \tau_{i-1}$ and $t_M = \tau_l$. Then there exist bounds $\delta^*$ and $d^*$ such that if $\overline{\delta} \leq \delta^*$ and $\overline{d} \leq d^*$ the function $W(\tau, t_0)$ admits its global maximizer $\hat{\tau}_i$ for $\hat{\tau}_i \in I = [\tau_i - \Delta\tau, \tau_i + \Delta\tau^+]$ where $I$ contracts to $\tau_i$ as $\overline{\delta} \to 0$ and $\overline{d} \to 0$. Moreover the estimated class $\hat{j}$ of the segment in $[t_0, \hat{\tau}_i]$ is equal the class of $i^{th}$ segment generated by system (6).*

The proof of this theorem relies on the following lemmas. In what follows we omit the dependence on $t_0$.

**Lemma 4.1** *Consider system (7). There exists $k_1 > 0$ such that*

$$\|(\hat{A}|\hat{b})(\tau) - (A_i|b_i)\|^2 \geq k_1 (\tau - \tau_i)^2 \ , \ \ \tau_i < \tau < \tau_{i+1}.$$

**Lemma 4.2** *Consider system (7) and $e_a(\tau)$ as defined in (10), there exists $k_2 > 0$ such that*

$$e_a(\tau) \geq k_2 \|(\hat{A}|\hat{b})(\tau) - (A_i|b_i)\|^2 \ , \qquad \tau_i < \tau < \tau_{i+1}.$$

**Lemma 4.3** *Let $A$ and $A_1$ be Hurwitz matrices and consider the pair of systems*

$$\dot{x} = Ax + b \tag{13}$$

$$\dot{z} = A_1 z + b_1 + d(t) \tag{14}$$

*with $x$ and $z$ in $\mathbb{R}^n$, $A, \ A_1 \in \mathbb{R}^{n \times n}$, $b$ and $b_1$ in $\mathbb{R}^n$, $\|d(t)\| \leq \overline{d}$ and $\|(A|b) - (A_1|b_1)\| \leq \overline{\delta}$. Then if $x(0) = z(0)$ there exist $k_3 > 0$ and $k_4 > 0$ such that*

$$\|x - z\|^2 \leq k_3 \overline{\delta} + k_4 \overline{d} \quad \forall t \geq 0. \tag{15}$$

**Lemma 4.4** *Let $e_p(\tau)$ and $e_a(\tau)$ denote parametric errors and approximation errors given in expressions (9) and (10) for the sequence of dynamical systems (6). Let $e_p^0(\tau)$ and $e_a^0(\tau)$ denote parametric errors and approximation errors for the related nominal system (7). Then there exist constants $k_p > 0$ and $k_a > 0$ such that*

$$e_p^0(\tau) - \Delta \leq e_p(\tau) \leq e_p^0(\tau) + \Delta \tag{16}$$

$$e_a^0(\tau) - \varepsilon \leq e_a(\tau) \leq e_a^0(\tau) + \varepsilon \tag{17}$$

*with $\Delta = k_p(\overline{d} + \overline{d}^2 + \overline{d}^3 + \overline{\delta} + \overline{\delta}^2 + \overline{\delta}^3)$ and $\varepsilon = k_a(\overline{d} + \overline{d}^2 + \overline{d}^3 + \overline{d}^4 + \overline{\delta} + \overline{\delta}^2 + \overline{\delta}^3 + \overline{\delta}^4 + \overline{\delta}^6)$.*

**Lemma 4.5** *Let the transition factor be given by (11) for system (6). There exist positive constants $c_1$ and $c_2$ such that if*

$$\Delta\tau = -c_1 \ln\left(\frac{1 - 2\beta}{1 - \beta}\right) \tag{18}$$

*then the transition factor is such that*

$$\mathrm{Tr}(\tau) \leq c_2 \beta, \quad \tau_{i-1} + \Delta\tau \leq \tau \leq \tau_i - \Delta\tau \tag{19}$$

$$\mathrm{Tr}(\tau) \geq \frac{1 - \rho_0 - \varphi}{2}, \quad \tau = \tau_i, \tag{20}$$

*for all $i$, where $\beta$ and $\varphi$ are perturbation dependent quantities and go to zero as the perturbation goes to zero.*

The proof of the theorem and of the lemmas can be found in [3, 5].

Note that the assumption that $t_0 = \tau_{i-1}$ is valid only at the first iteration in which $t_0 = \tau_0$. Then we find the maximizer $\hat{\tau}_1$ of $W(\tau)$ for $\tau \in (\tau_0, \tau_l)$ which lies in an interval $I = [\tau_1 - \Delta\tau, \tau_1 + \Delta\tau^+]$ around $\tau_1$ and is an estimate of the first switching time $\tau_1$. Then we have to set $t_0$ for the second iteration so that the first switching point encountered after $t_0$ is $\tau_2$. In order to do this we set $t_0 = \hat{\tau}_1 + \Delta\tau$ so that we make sure that the first switching time encountered is $\tau_2$ and not $\tau_1$ again. In fact if the maximization process of $W$ takes place with $t_0 > \tau_i$ and in the worst case scenario with $t_0 = \tau_i + \Delta\tau + \Delta\tau^+$ nothing changes as long as $T - (\Delta\tau + \Delta\tau^+) > 2\Delta\tau$ that implies an other condition on the noise level, which added to the ones found in Theorem 4.1 give new values for $d^*$ and $\delta^*$.

**Remark 4.1** Assume that in the expression of $W(\tau)$ given in (12) we add a factor $s(\tau)$ with the properties that $s(\tau) \in [\frac{1}{K}, 1)$ for all $\tau$, $K \geq 1$ and $s(\tau) \geq 1 - \nu$ for $\tau_{i-1} < \tau \leq \tau_i$, with $\nu \ll 1$. Then the proof of Theorem 4.1 proceeds at the same way with minor modifications (see [3] for details.)

## 5 Algorithm Implementation

The segmentation algorithm was implemented in MATLAB 6.0 in the case of planar motion modeled by the discrete time version of

$$\dot{X} = \begin{pmatrix} A_x & 0 \\ 0 & A_y \end{pmatrix} X + b, \quad A_{x,y} = \begin{pmatrix} 0 & 1 \\ a_{1,x,y} & a_{2,x,y} \end{pmatrix} \tag{21}$$

where $X = (x, \dot{x}, y, \dot{y})^T$, with $xy$ coordinates in the plane, and $b = (0, b_x, 0, b_y)^T$. The interconnection condition that holds in this case is by replacing $\dot{x}$ in equation (8), with $(\dot{x}, \dot{y})^T$ (which does not affect result of Lemma 4.5.) The function $W(\tau)$ takes the form

$$W(\tau) = e^{\left(-\frac{(e_a(\tau)-e_a^c)^2}{\sigma_a^2}\right)} \frac{\mathrm{Tr}(\tau) e^{(-\bar{e}_p^2(\tau))} s(\tau) p(\tau)}{a + e_a(\tau)} \tag{22}$$

The term $\exp(-\frac{(e_a(\tau)-e_a^c)^2}{\sigma_a^2})$ represents a Gaussian distribution of the approximation error around a mean value that can be obtained by processing part of the data. The parametric error $\overline{e}_p$ takes into account also possible non-spherical shapes of the distribution of the parameters around the centers. Using the same notation as used for defining $e_p$ it can be written as $\overline{e}_p^2(\tau) = \min_j (\hat{A} - A_j)^T \Sigma_j^{-1} (\hat{A} - A_j) / \sqrt{\det(\Sigma_j)}$. The term $s(\tau)$ satisfies the properties described in Remark 4.1, which can be used to include additional information other than that derived from the dynamical parameters. Its choice will be described in the experiment section. Since pauses occur for the drawing tasks described in the next section and must be taken into account by the algorithm, we introduce $p(\tau) = k/(\text{pause length})$. Such a term penalizes segments containing pauses, which are not supposed to take place in a moveme.

The time $t_0$ that is the starting point of each iteration is obtained as explained at the end of the proof of Theorem 4.1. The way we implement this is by taking into account that the end of each segment reaches a steady state with poor amount of signal. We estimate the length of the signal after $\hat{\tau}_i$ that has a poor content of information: this gives an estimate of the time interval we have to add to $\hat{\tau}_i$ in order to find a point $t_0$ which lies in the following segment.

The segmentation algorithm can be summarized as:

(i) initialization: $t_0 = \tau_0$, $t_M = \tau_l$, $i = 1$;

(ii) maximize $W(\tau)$ for $\tau \in (t_0, t_M]$: $\hat{\tau}_i = \max_{\tau \in (t_0, t_M]} W(\tau)$;

(iii) compute class $j$ of the segment found: $j = \arg_{k \in \{1,\dots,m\}} (\|\hat{A}(\hat{\tau}_i) - A_c^k\|) \leq r_k$;

(iv) compute $\Delta\tau$;

(v) $t_0 = \hat{\tau}_i + \Delta\tau$;

(vi) $i = i + 1$;

(vii) go to (ii);

where we recall that $\tau_0$ and $\tau_l$ are the starting and ending points of the data stream. Note that the number $l$ of segmenting points does not need to be known *a priori*, only $t_l = t_M$ is supposed to be known.

## 6 Experimental Results

To test our approach, we studied a 2D drawing task in which a set of shapes were drawn by five different subjects using a computer mouse.

### 6.1 Experimental setup

Our subjects drew using the XPaint program on a PC running Red Hat Linux 7.2 with a screen measuring $1600 \times 1200$ pixels and a working window of $700 \times 500$ pixels. The user left the trace of the trajectory in the working window only when the left mouse button was pressed. For acquiring $x$ and $y$ time traces we implemented a C routine which was activated in the background at the beginning of each experimental session and sampled the $(x, y)$ position of the pointer everywhere on the screen at the rate of 100 Hz and a spatial resolution of one pixel. The routine makes use of XWindow libraries and captures the pointer position through the function "XQueryPointer" which is called by a timer every 10 ms and gives the coordinates in pixels with respect to the upper left corner of the screen. Every 30 minutes the data was saved into files by means of a parallel process. The data so obtained consists of an array with three columns containing time, $x$ position at that time, $y$ position at the same time. The time interval between one sample and the following one turned out to be mostly constant except for slight variations every once in a while due to higher priority of other processes. In order to have constant sampling time the data was processed through an algorithm that linearly interpolates data in the regions in which the time interval is not exactly 10 ms. Pixelization of the coordinates does not heavily affect the data since the trajectories under study are usually more than 50 pixels long.

We defined 4 different drawings by means of prototypes: car, sun, ship, and house. Each of the 5 subjects was shown the prototypes and was asked to reproduce them on a $700 \times 500$ pixel canvas; the dimensions of each drawing could be chosen arbitrarily according to the ones with which the user was more comfortable, the only specification was to reproduce the prototypes with as high fidelity as possible in a reasonable amount of time. Each subject drew 10-20 examples for each shape. In order to accomplish each drawing task

the user had to perform a sequence of actions such as "reach a point A" and "draw a line up to point B". These actions are the ones that we will consider as candidates for being elementary motions and then defining a pair of movemes. The idea is then to use the result of Theorem 4.1 so as to find the sequence of reach and draw movements that the user did in order to accomplish the task and the switching times between one and the other.

### 6.2 Classification

To obtain reach and draw dynamical parameters according to model (21), 140 examples of reach trajectories were captured from a video game implemented in MATLAB 6.0, and 140 examples of draw trajectories were segmented out from cars and houses of 2 of the subjects. By proceeding with standard pattern recognition techniques (see [1] for example), we trained a Gaussian classifier on this data. Since our data set contains also circular shapes like the wheels of the cars, we also introduced a circle class beyond the reach and draw classes. The dynamical model by which we represent such a class is the coupled version of system (21):

$$\dot{X} = \begin{pmatrix} A_x & C_x \\ C_y & A_y \end{pmatrix} X + b, \quad C_{x,y} = \begin{pmatrix} 0 & 0 \\ c_{1,x,y} & c_{2,x,y} \end{pmatrix} \tag{23}$$

so that we have 8 parameters for classification. We considered an additional parameter that is the value of $\omega/T$ where $\omega$ is the principal frequency estimated and $T$ is the duration of the trajectory: we expect for a circle that to be about $2\pi$. We then trained a Gaussian classifier in $\mathbb{R}^9$ on a training set composed of 101 examples derived from the wheels of the cars. The cumulative training and testing errors for the three-class classification problem are respectively 3.4% and 4.6%, when testing was performed on a test set (deriving from different subjects) of 323 additional reach examples, 118 additional draw examples, and 124 examples of the wheels of the cars and suns of two other subjects. These classification errors increase if we choose dynamical models different from model (21) for reach and draw: we tried first, second, third and fourth order coupled and decoupled dynamical systems, and in all the cases the classification performance was worse. This justifies the choice of (21) to represent reach and draw movemes.

### 6.3 Segmentation algorithm performance

We implemented the proposed segmentation algorithm in MATLAB on the data acquired as described in the previous sections considering a number of three movemes: the reach, the draw, and the circle movemes.

For improving the performance we introduced in the expression of $W$ given by (22) the term $s(\tau)$, which takes explicitly $xy$ coupling information into account. The need for introducing this term comes from the fact

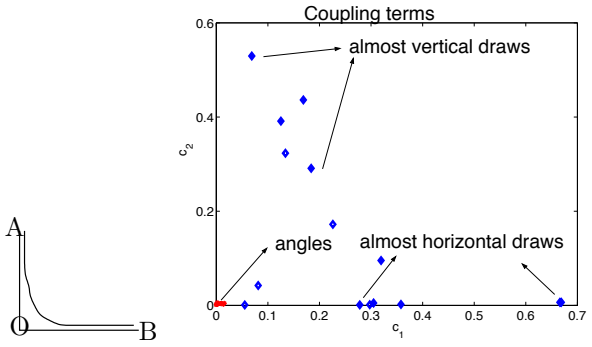


**Figure 2:** The coupling parameters (diamonds) obtained for the vertical and horizontal draws separately and the coupling parameters (stars) obtained for angles AOB.

that system (21), chosen for representing the movemes, can approximate with acceptable approximation errors angles (shown as $AOB$ in Figure 2) in $xy$ plane, while having parameters that are still classified as reach or draw. Then the estimated parameters of system (21) for a given trajectory do not contain information to discriminate between one draw and an angle, and this is due to the absence of any $xy$ coupling information. Such information would discriminate quite clearly between the single draw case and the angle of the kind of AOB shown in Figure 2: for approximating an $xy$ trajectory with the simplest system containing $xy$ coupling, such as

$$\begin{pmatrix} \dot{x} \\ \dot{y} \end{pmatrix} = \begin{pmatrix} d_1 & c_1 \\ c_2 & d_2 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + b, \tag{24}$$

we obtain estimated coupling terms $(\hat{c}_1, \hat{c}_2)^T = \hat{c}$ that are close to zero for the angle and bounded away from zero for the single draw, as shown in Figure 2. Thus we choose a shaping term in the expression (22), for the reach and the draw classes, of the form

$$s(\tau) = \frac{1}{1 + L\ \exp(-(\hat{c} - \overline{c})^T \Sigma_c^{-1} (\hat{c} - \overline{c}))/\sqrt{\det(\Sigma_c)}},$$

with $L \geq 1$, where $\overline{c}$ and $\Sigma_c$ are obtained by means of a learning phase in which we train the Gaussian classifier, $\exp(-(z-\overline{c})^T \Sigma_c^{-1}(z-\overline{c}))/\sqrt{\det(\Sigma_c)}$, on a set of about 25 examples of angles. The value of $\Sigma_c$ turns out to be very small resulting in a very narrow Gaussian around the mean as we can deduce form the concentrated cluster of angle's parameters of Figure 2. Thus angles will be penalized with a value of $s(\tau) << 1$ for $L$ big. By simple computation we can show that $s(\tau)$ satisfies the conditions of Remark 4.1.

Since in our data set some squares (windows of the houses) have rounded angles and look very similar to circles, we obtained a slight performance improvement by introducing a higher level step in the algorithm, in

which we decide if a segment detected as a circle is more likely to be a square. At each iteration in which a circle is detected, to decide if the data segmented as a circle is more likely to be a square, we run the segmentation algorithm again on that data without the circle classifier (that is by assuming that the data is a sequence of reaches or draws or both). Then if the algorithm segments it into a sequence of draws, we compute the likelihood of each draw that has been detected as the product $\exp\left(-\frac{(e_a(\tau)-e_a^c)^2}{\sigma_a^2}\right)\exp(-\bar{e}_p^2(\tau))$, which is the part of (22) that quantifies how good the detected segment is as representative of its class. We then average the likelihood of all the detected draws and compare it to a threshold obtained by preprocessing some of the squares and some of the circles (about 10 examples each). This higher level process does not affect performance drastically, but turns out to be helpful in 3–4 cases in which the windows of the houses have not evident corners. For minimizing the algorithmic time, we set $t_M$ *a priori* to be $t_0$ plus the maximum duration of a segment that in our case turned out to be 500 time steps.

The algorithm takes as input the signal $(x(t), y(t))$ and gives as outputs a sequence of segmentation points and the classification of the trajectory between two detected segmentation points. The algorithm performance was computed by assuming a ground truth: we expected to detect a segmentation point at the beginning and at the end of each moveme and also we expected each one of them to be properly classified. Then the algorithm error was computed as the sum of classification error (i.e., a trajectory which is correctly segmented but wrongly classified) and segmentation error (i.e., a trajectory which is over segmented, or a missed segmentation point). An estimate of such an error was computed on segmentation results on cars, ships, houses sequences deriving from two subjects each. The error estimate is reported in Table 1, which was obtained by counting the total number of segmentation points detected (denominator) and the number of segments that were clearly mis-classified or mis-segmented (numerator).

Table 1: Algorithm error

| | class. error | segm. error | cum. error |
|---|---|---|---|
| CAR | $\frac{112}{1333} = 8.4\%$ | $\frac{20}{1333} = 1.5\%$ | 9.9% |
| HOUSE | $\frac{108}{1050} = 10.29\%$ | $\frac{23}{1050} = 2.19\%$ | 12.48% |
| SHIP | $\frac{99}{1093} = 9.06\%$ | $\frac{3}{1093} = 0.27\%$ | 9.3% |

The average error is about 10.5%. We report some pictures (Figures 3, 4, 5), which show the segments classified as reach, the segments classified as draws, the ones classified as circles and the unclassified ones. The little circles represent the segmentation points that the algorithm found.

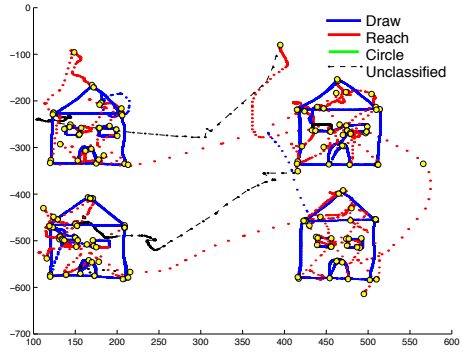


**Figure 3:** Segmentation results on 4 houses of subject 3.

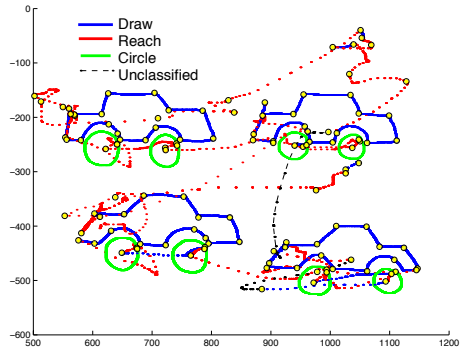


**Figure 4:** Segmentation results on 4 cars of subject 1.

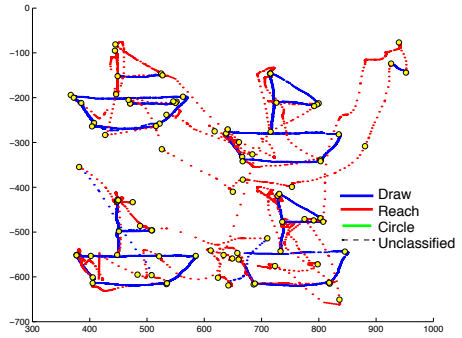


**Figure 5:** Segmentation results on 4 ships of subject 1.

### 6.4 Categorization

To illustrate a possible usage of the output of the segmentation algorithm, we want to recognize what is the category (car, house, ship) of a certain drawing based on the number of reach, draw and circles that according to the segmentation algorithm composes it. To this

aim we associate to each sequence corresponding to one of the three shapes the vector of natural numbers $(R, D, C)^T$, which are the number of reaches, draws and circles detected. We then train a Gaussian classifier with the $(R, D, C)^T$ vectors of 15 cars (as drawn by one subject), 9 houses (from two subjects), and 4 ships (from two subjects), and obtain 0% training error. The test is performed on the remaining 53 ships (from three subjects), 36 houses (from three subjects), 31 cars (from two different subjects) and we obtain 5.4% test error, which is quite small since we based our discrimination just on the basis of the number of movemes and not on their order.

## 7 Conclusions

We have addressed the classification and segmentation problems and proposed an algorithm with error analysis. The experimental results show that the segmentation and classification performance of the proposed algorithm is about 90% on our data set. We finally show, with an example, that the output of the segmentation algorithm can be used to solve higher level tasks like discriminating between activities composed by movemes and found an error on our data set of about 5% when using a simple-minded recognition strategy.

Future directions include the exploration of 3D motion of the human body, generalization of the current algorithms to the on-line case, and set up a possible solution to the prediction problem. We are also interested in exploring the use of dynamical event constraints to improve the quality and robustness of segmentation and classification algorithms. For example we know that in the sequence "step-step-reach-lift" for answering the phone, it is not possible to lift the phone before having reached it. These kinds of constraints could be embedded in a model which gives a structure to the way in which movemes are composed. Future theory directions include the observability and estimation problems of hybrid systems.

## 8 Acknowledgments

This project has been funded in part by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726). The authors would like to acknowledge the reviewers for providing information about related work.

## References

[1] C.M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon, Oxford, 1995.

[2] C. Bregler and J. Malik. Learning and recognizing human dynamics in video sequences. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 568–674, Puerto Rico, 1997.

[3] D. DelVecchio, R. M. Murray, and P. Perona. Decomposition of human motion into dynamic based primitives with application to drawing tasks. Technical report, Caltech, report number 2002-004, URL: `http://www.cds.caltech.edu/ murray/papers/ index.shtml`, aug 2002.

[4] D. DelVecchio, R. M. Murray, and P. Perona. Primitives for human motion: a dynamical approach. In *Proceedings of the 2002 IFAC 15th World Congress*, Barcelona, Spain, 2002.

[5] D. DelVecchio, R. M. Murray, and P. Perona. Decomposition od human motion into dynamic based primitives with application to drawing tasks. *Automatica (submitted)*, 2003.

[6] J.W. Fisher, A.T.Ihler, and P.A. Viola. Learning informative statistics: A nonparametric approach. In *Conf. on Neural Information Processing Systems*, Marriott City Center Hotel in Denver, 1999.

[7] L. Goncalves, E. Di Bernardo, and P. Perona. Reach out and touch space (motion learning). In *Proc. of the Third International Conference on Automatic Face and Gesture Recognition*, pages 234–239, Nara, Japan, April 14-16 1998.

[8] F. Gustafsson. *Adaptive Filtering and Change Detection*. John Wiley & Sons, 2000.

[9] M. Lavielle. Optimal segmentation of random processes. *IEEE Trans. on Signal Processing*, 46:1365–1373, May 1998.

[10] L. Ljung. *System Identification*. Prentice Hall, New Jersey, 1999.

[11] C. Lu, H. Liu, and N.J. Ferrier. Multidimensional motion segmentation and identification. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 629–636, Hilton Head Island, South Carolina, 2000.

[12] D. Ormoneit, T. Hastie, and M.J. Black. Functional analysis of human motion data. In *Proc. 5th World Congress of the Bernoulli Society for Probability and Mathematical Statistics and 63rd Annual Meeting of the Institute of Mathematical Statistics*, Guanajuato, Mexico, 2000.

[13] V. Pavlovic and James M. Rehg. Impact of dynamic model learning on classification of human motion. In *IEEE Conf. Computer Vision and Pattern Recognition*, Hilton Head Island, 2000.

[14] T. Söderström and P. Stoica. *System Identification*. Prentice Hall. Hemel Hempstead, 1989.

[15] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995.