Submitted, 2003 European Control Conference
http://www.cds.caltech.edu/~murray/papers/2002n_dmp03-ecc.html

CLASSIFICATION OF HUMAN ACTIONS 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: We develop the study of primitives of human motion, which we refer to as *movemes*. The idea is to understand human motion by decomposing it into a sequence of elementary building blocks that belong to a known alphabet of dynamical systems. Where do these dynamic primitives come from in practice? How can we construct an alphabet of movemes from human data? In this paper we address these issues. We define conditions under which collections of signals are well-posed according to a dynamical model class M and thus can generate movemes. Using examples from human drawing data, we show that the definition of well-posedness can be applied in practice so to establish if sets of actions, reviewed as signals in time, can define movemes.

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. With the development of information technology we can expect that computer systems will be increasingly embedded in our environment, so that human-machine interaction will need interfaces that are easier to use and more natural. In particular the possibility of interacting with computerized environment without the need for special external equipment is attractive. Several works (see for example (Laptev and Lindeberg, 2001; Waldherr *et al.*, 1998) and the earlier work on building human-machine interfaces using vision (Goncalves *et al.*, 1995; Munich and Perona, 1996; Wilson and Bobick, 1995; Yacoob and Davis, 1996; Wellner, 1991)) ask the question of whether it is possible to develop computerized equipment able to communicate with humans in similar way. As described extensively in (Collins *et al.*, 2000) there is also an immediate need for automated surveillance systems in commercial, law enforcement, and military applications. Surveillance cameras are present in banks, stores, and parking lots; it is desirable to develop continuous automated monitoring to alert security officers about suspicious human activity while there is still time to prevent a possible crime. Other applications include videogames and animation where virtual human motion is based on the learning and description of real human motion (see for example (Zordan and Hodgins, 1999) and (Silva *et al.*, 1997)). Another important application is biomechanics (see for example (Pedotti *et al.*, 1989)).

A fundamental problem in detecting and recognizing human action is one of representation. As explained in (DelVecchio *et al.*, 2002), our point of view is that human activity should be decomposed into building blocks which belong to an "alphabet" of elementary actions. We refer to these primitives of motion as *movemes*. We thus aim 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 (Bregler and Malik, 1997). 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. (Goncalves *et al.*, 1998) studied motions that were parametrized by an initial condi-

tion and a target. They proposed that movemes ought to be parametrized by goal and style parameters. Their moveme models are phenomenological and non-causal.

What is the alphabet of movemes? Which are the dynamical models that we should use to represent them? Where do movemes come from in practice? When human actions can define movemes according to a dynamical model class M? To answer these questions we use system identification tools (Ljung, 1999; S*ö*derstr*ö*m and Stoica, 1989), we recall the formal definition of moveme already given in (DelVecchio *et al.*, 2002), and we introduce the classification problem as a standard problem of pattern recognition (Bishop, 1995; Vapnik, 1995) in a suitable space. In (DelVecchio *et al.*, 2002) and (DelVecchio *et al.*, n.d.) some classification results were presented because they were needed for setting the solution of the segmentation problem. However the assumption that the actions considered were defining movemes was tacitly made. In this paper we show with examples from real data that such assumption may not hold in practice and explain why this may happen. We thus propose a way to establish when real data (seen as signals in time) can allow the definition of a set of movemes. From this study we derive the definition of well-posed sets of signals as signals that can define movemes, and we show how such a definition can be checked in practice. In the case of computer drawing data we find two sets of actions that are well-posed and define the "reach" and the "draw" movemes. In the same dataset we find also a class of motions that we refer to as "free motion" for which the well-posedness test fails, and therefore it cannot define a moveme in the already constructed reach-draw moveme alphabet.

2. DYNAMICAL DEFINITION OF MOVEME

We recall in this section a relaxed version of the definition of moveme already presented in (DelVecchio *et al.*, 2002), 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

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

for all $t \geq t_0$, 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 $\mathcal{M} = \{M^1, ..., M^l\}$ of mutually dynamically independent model sets is called a *moveme*. It is clear from the definition that in order to establish if a model set is a moveme we should have at least an other model set so to be able to check property (ii) of the above definition. Then the notion of a model set of being a moveme is relative to a context comprising other model sets.

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 $\Upsilon(A|b) = A$. For such a class of models we make the following assumption.

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 $v \in \mathbb{R}^{n+1}$,

$$v^T\overline{x}(t) = 0,\ t \in [t_1, t_2],\ t_2 > t_1 \quad \Longrightarrow \quad v = 0,$$

where $\overline{x} = (x^T, 1)^T$.

This assumption means that the description that model (1) provides for $x(t)$ is minimal in the sense that $x(t)$ cannot also be described by a lower order dynamical system. In fact if $v^T\overline{x}(t) = 0,\ t \in [t_1, t_2],\ t_2 > t_1$ for some $v \neq 0$ then $x_n(t) = \alpha_0 + \alpha_1 x_i(t) + ... + \alpha_{n-1}x_{n-1}(t)$ for any t, therefore the dynamics can be described just in terms of $x_1(t), ..., x_{n-1}(t)$ and $x_n(t)$ can be derived algebraically. A direct consequence of such an assumption is that we have a one-one correspondence between $x(t)$ and parameters $(A|b)$ of model (1), so that we have the following lemma.

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

$$\begin{aligned} \dot{x} &= A_1x + b_1 \\ \dot{z} &= A_2z + b_2 \end{aligned}$$

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

Proof ($\Leftarrow$) If $(A_1|b_1) = (A_2|b_2)$ then $z(t) = x(t)$ for all t by uniqueness of solutions.
($\Rightarrow$) If $z(t) = x(t)$ for all t then $\dot{z}(t) = \dot{x}(t)$ for all t, so that $A_1 x + b_1 = A_2 z + b_2$. This implies $[(A_1|b_1) - (A_2|b_2)]\overline{x}(t) = 0$ for all t, which by Assumption 2.1 (applied to each column) implies $(A_1|b_1) = (A_2|b_2)$. $\square$
This lemma shows that property (i) of Definition 2.1 is satisfied by our choice of M and $\mathcal{U}$. Property (ii) is verified by choosing 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:

$$\begin{aligned} &\mathcal{C}^j = B_{r_j}(A_c^j)\,, && j = 1, ..., m \\ &\mathcal{C}^j \bigcap \mathcal{C}^k = \{\emptyset\}, && j \neq k \end{aligned} \tag{2}$$

where m is the number of movemes and the matrix norm is the Frobenius norm. In what follows we assume that the sets $\mathcal{C}^j$ are described by equation (2). Then we have constructed a set $\mathcal{M} = \{M^1, ..., M^m\}$ of m movemes where $M^k = \{M((A|b))|A \in \mathcal{C}^k\}$, for $k \in \{1, ..., m\}$ and M is in the form given by equation (1).

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 (Ljung, 1999)):

$$(\hat{A}|\hat{b}) = \arg\min_{(A|b)} \frac{1}{2} \int_{t_0}^{T} (\dot{x} - (A|b)\overline{x})^T (\dot{x} - (A|b)\overline{x}) dt \tag{3}$$

with $\overline{x} = (x^T, 1)^T$, which gives the least squares estimate of parameters $(\hat{A}|\hat{b})$ so to get the estimate of x in model class (1) as

$$\dot{\hat{x}} = \hat{A}\hat{x} + \hat{b}, \qquad \hat{x}(t_0) = x(t_0).$$

In the case in which $x(t)$ has been generated by (1), by virtue of Assumption 2.1 it is easy to check that (3) 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 $\hat{A} \in \mathcal{C}^k$. This is equivalent by virtue of (2) to finding $k \in \{1, .., j, ..m\}$ such that $\|\hat{A} - A_c^k\| \leq r_k$, whose solution is unique since the sets $\mathcal{C}^k$ are all not intersecting. Then

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

The following section addresses the same classification problem in a more general situation in which $x(t)$ has been generated by a perturbed version of system (1).

2.2 *Classification Problem*

Let the signal $x(t)$ be generated by the perturbed version of system (1):

$$\begin{aligned} \dot{x} &= Ax + b + d(t) \\ y &= x \end{aligned} \tag{4}$$

with $A \in \mathcal{C}^j$, 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? Since $A \in \mathcal{C}^j$, there exists $\delta < r_j$ such that $A = A_c^j + \delta U$ with U a unit norm matrix and A_c^j center of $\mathcal{C}^j$. Then system (4) becomes

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

Then the problem of classifying $x(t)$ as output of moveme M^j becomes the same as identifying j in system (5) for some conditions on δ and $d(t)$. In the previous section we showed that if $d(t) = 0$ then we can exactly identify $A_c^j + \delta U$ and then correctly classify $x(t)$. The presence of $d(t)$ induces an estimation error so that $\hat{A}$ will not be equal to $A_c^j + \delta U$, but it is not necessary to achieve equality for our purpose as the following lemma shows.

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 (2). Let $\hat{A}$ be the least squares estimate according to (3). There exist positive constants $\overline{d}$ and $\overline{\delta}$ such that if $\delta \leq \overline{\delta}$ and $\|d(t)\| \leq \overline{d}$, then

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

Proof. By equation (3) we have

$$(\hat{A}|\hat{b}) = \left(\int_{t_0}^{T} \dot{x}(t)\ \overline{x}(t)^T dt\right)\left(\int_{t_0}^{T} \overline{x}(t)\ \overline{x}(t)^T dt\right)^{-1}$$

where we can invert $\left(\int_{t_0}^{T} \overline{x}(t)\ \overline{x}(t)^T\right)$ if either $d(t) = 0$ by Assumption 2.1, or $d(t) \neq 0$ by the fact that $d(t)$ is realization of white noise that is uncorrelated in time. Using equation (5), this expression becomes

$$\begin{aligned} (\hat{A}|\hat{b}) &= (A_c^j + \delta U|b) \\ &\quad + \left(\int_{t_0}^{T} d(t)\overline{x}(t)^T dt\right)\left(\int_{t_0}^{T} \overline{x}(t)\ \overline{x}(t)^T dt\right)^{-1} \end{aligned}$$

which leads with some algebra to

$$\|\hat{A} - A_c^j\| \leq \|(\hat{A}|\hat{b}) - (A_c^j|b)\| \leq \overline{\delta} + \overline{d}\ c$$

where $\overline{\delta}$ and $\overline{d}$ are upper bounds on δ and $d(t)$, and c is a suitable positive constant which exists since $x(t)$ is bounded by the stability properties of the dynamics. Then in order for $\|\hat{A} - A_c^k\| \leq r_k$ to hold for $k = j$ it is sufficient that

$$\|\hat{A} - A_c^j\| \leq \overline{\delta} + \overline{d}\ c \leq r_j \tag{6}$$

which is verified if, for example, $\overline{\delta} = r_j/2$ and $\overline{d} = r_j/(2c)$, which give upper bounds on δ and $d(t)$. Note that the uniqueness of the solution for k comes from the fact that the sets $\mathcal{C}^k$, $\mathcal{C}^j$ for $k \neq j$ are not intersecting as equation (2) guarantees. If such

a requirement is not satisfied even when equation (6) holds, then the solution $k \in \{1, ...j, ...m\}$ of $\|\hat{A} - A_c^k\| \leq r_k$ may not be unique, leading to ambiguity in the classification. □

2.3 *Well-posedness*

As the previous section highlighted, the basic requirement for solving the classification problem is the one of having non intersecting sets in parameter space characterizing the sets of dynamical models M^j, $j = 1, ..., m$. In practice the sets $\mathcal{C}^j$ and $\mathcal{C}^k$, $j \neq k$ may be not defined *a priori* but are derived from finite sets of signals $\mathcal{S}^j$ and $\mathcal{S}^k$, whose characteristics make each element of one set different from each element of the other and therefore we can say that they define two classes of signals. When can these two classes of signals define two movemes M^j, M^k according to Definition 2.1? Let the two classes $\mathcal{S}^j$ and $\mathcal{S}^k$ be composed by signals $s_i^j(t) = Y(M(\Theta_i^j)|_{x_{0i}^j, u_i^j})(t)$, for $s_i^j(t) \in \mathcal{S}^j$, and $s_i^k(t) = Y(M(\Theta_i^k)|_{x_{0i}^k, u_i^k})(t)$, for $s_i^k(t) \in \mathcal{S}^k$. Let $\mathcal{F}_M$ be an estimation procedure establishing a one to one mapping between the signal $Y(M(\Theta)|_{x_0,u})(t)$ and the couple (Θ, u) which exists by virtue of (i) of Definition 2.1. Then we have

$$\begin{aligned} (\Theta_i^k, u_i^k) &= \mathcal{F}_M(s_i^k(t)) \qquad & s_i^k(t) \in \mathcal{S}^k \\ (\Theta_i^j, u_i^j) &= \mathcal{F}_M(s_i^j(t)) \qquad & s_i^j(t) \in \mathcal{S}^j. \end{aligned}$$

Let $f_s : (E \times \mathcal{U}) \to E$ be the selection operator, such that $f_s(\Theta, u) = \Theta$, which selects the first element of the couple (Θ, u). Then define $f_M := \Upsilon \circ f_s \circ \mathcal{F}_M$, which associates to each signal $s(t)$ the corresponding parameter θ lying in $E' \subset E$. We can then write that $\mathcal{C}^j$ is the image of $\mathcal{S}^j$ through f_M and the same for $\mathcal{C}^k$:

$$\begin{aligned} f_M(\mathcal{S}^j) &= \mathcal{C}^j \\ f_M(\mathcal{S}^k) &= \mathcal{C}^k. \end{aligned} \tag{7}$$

Definition 2.2. Classes of signals $\mathcal{S}^j$ and $\mathcal{S}^k$ with elements $s_i^j(t) = Y(M(\Theta_i^j)|_{x_{0i}^j, u_i^j})(t)$, for $s_i^j(t) \in \mathcal{S}^j$, and $s_i^k(t) = Y(M(\Theta_i^k)|_{x_{0i}^k, u_i^k})(t)$, for $s_i^k(t) \in \mathcal{S}^k$, such that the corresponding sets $\mathcal{C}^j$ and $\mathcal{C}^k$ given in (7) are non intersecting, that is $\mathcal{C}^j \bigcap \mathcal{C}^k = \{\emptyset\}$, are said to be *well-posed* classes according to model M.

From this definition it follows immediately that well-posed classes of signals define movemes according to Definition 2.1. In practice we have access to a finite set of signals, $S^j = \{s_1^j(t), ..., s_{n_j}^j(t)\}$ and $S^k = \{s_1^k(t), ..., s_{n_k}^k(t)\}$, which belong to the two classes $\mathcal{S}^j$ and $\mathcal{S}^k$, with $s_i^j(t) = Y(M(\Theta_i^j)|_{x_{0i}^j, u_i^j})(t)$ for $i \in \{1, ..., n_j\}$ and $s_i^k(t) = Y(M(\Theta_i^k)|_{x_{0i}^k, u_i^k})(t)$ for $i \in \{1, ..., n_k\}$. Let $\hat{\mathcal{C}}^j$ and $\hat{\mathcal{C}}^k$ be the images, through f_M, of the sets S^j and S^k respectively. By construction we have $\hat{\mathcal{C}}^k \subset \mathcal{C}^k$ and $\hat{\mathcal{C}}^j \subset \mathcal{C}^j$, so that potentially we can have trivial intersection between $\hat{\mathcal{C}}^j$ and $\hat{\mathcal{C}}^k$, and a no-empty intersection between the sets $\mathcal{C}^k$ and $\mathcal{C}^j$. This creates a problem since if we check Definition 2.2 with $\hat{\mathcal{C}}^j$ and $\hat{\mathcal{C}}^k$, which are the only ones to which we have access, the classes of signals $\mathcal{S}^j$ and $\mathcal{S}^k$ turn out to be well-posed. The situation is

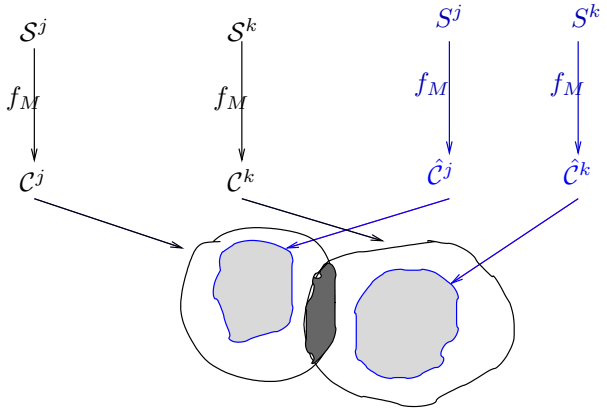

Fig. 1. Relation between sets $\hat{\mathcal{C}}^j$ and $\hat{\mathcal{C}}^k$ and $\mathcal{C}^j$ and $\mathcal{C}^k$.

depicted in Figure 1. The issue comes from the fact that we will use the light sets ($\hat{\mathcal{C}}^j$ and $\hat{\mathcal{C}}^k$) for solving the classification problem ignoring the existence of the dark region that is generating signals with undecidable class. Then, one needs to check if $\mathcal{S}^k$ and $\mathcal{S}^j$ are well-posed. The following lemma gives a possible way to check for well-posedness without knowing the sets $\mathcal{C}^j$ and $\mathcal{C}^k$.

Lemma 2.3. Let $y(t) = Y(M(\Theta)|_{u,x_0})(t)$ denote the output of model M for a choice of Θ, u and x_0. Assume to fix u, x_0 and $\Theta|_{E-E'}$, so that $y(t) = Y(M(\theta))(t)$, and let the classes $\mathcal{S}^j$ and $\mathcal{S}^k$ be defined as

$$\begin{aligned} \mathcal{S}^j = \{y(t) | y(t) &= Y(M(\theta))(t) \\ &\text{and } g_j(y, \dot{y}, t) = 0,\ h_j(y, \dot{y}, t) \leq 0\} \end{aligned}$$

$$\begin{aligned} \mathcal{S}^k = \{y(t) | y(t) &= Y(M(\theta))(t) \\ &\text{and } g_k(y, \dot{y}, t) = 0,\ h_k(y, \dot{y}, t) \leq 0\} \end{aligned}$$

for some functions g_j, g_k, h_j and h_k. Then the classes of signals $\mathcal{S}^j$ and $\mathcal{S}^k$ are well-posed if and only if the system

$$\begin{aligned} y(t) &= Y(M(\theta))(t) \\ g_j(y, \dot{y}, t) &= 0 \\ h_j(y, \dot{y}, t) &\leq 0 \\ g_k(y, \dot{y}, t) &= 0 \\ h_k(y, \dot{y}, t) &\leq 0 \end{aligned} \tag{8}$$

is infeasible.

Proof ($\Rightarrow$). Let us show that well-posed classes $\mathcal{S}^j$ and $\mathcal{S}^k$ imply infeasibility of (8). According to Definition 2.2 this is equivalent to showing that non-intersecting sets $\mathcal{C}^j$ and $\mathcal{C}^k$ (defined in equation (7)) imply infeasibility of the system of equations (8). Let again $\mathcal{F}_M$ be the one to one mapping between the signal

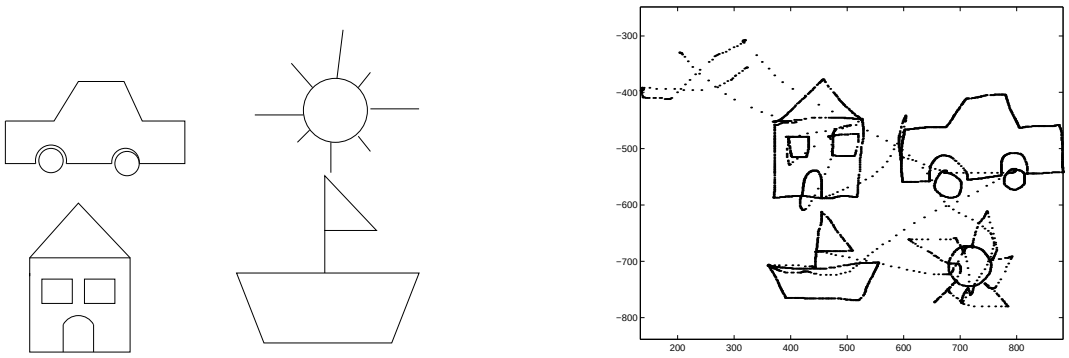

Fig. 2. Prototypes of the four shapes shown to the subject and example of traces in xy plane captured by the capturing system.

$Y(M(\Theta)|_{x_0,u})(t)$ and the couple (Θ, u) which exists by virtue of (i) of Definition 2.1, and since input u, initial condition x_0 and $\Theta|_{E-E'}$ have been fixed, $\mathcal{F}_M$ becomes one to one correspondence between $y(t) = Y(M(\theta))(t)$ and θ. Then we can redefine the sets $\mathcal{C}^j$ and $\mathcal{C}^k$ as

$$\mathcal{C}^j = \{\theta | \theta = \mathcal{F}_M(y(t)), \text{ and } y(t) \in \mathcal{S}^j\} \quad (9)$$

and

$$\mathcal{C}^k = \{\theta | \theta = \mathcal{F}_M(y(t)), \text{ and } y(t) \in \mathcal{S}^k\}. \quad (10)$$

If (8) is feasible then there exist $y(t)$ such that $y(t) \in \mathcal{S}^j$ and $y(t) \in \mathcal{S}^k$ and also there exist $\theta^* : \ y(t) = M(\theta^*)(t)$, so that by (9) and (10) $\theta^* \in \mathcal{C}^j$ and $\theta^* \in \mathcal{C}^k$, which in turn implies $\mathcal{C}^k \bigcap \mathcal{C}^j \neq \{\emptyset\}$. Then we have shown that trivial intersection of sets $\mathcal{C}^j$ and $\mathcal{C}^k$ defined in (9) and (10) implies infeasibility of (8). ($\Leftarrow$). Let us show now that if classes $\mathcal{S}^j$ and $\mathcal{S}^k$ are not well-posed, then system (8) is feasible. By Definition 2.2, this is equivalent to show that if $\mathcal{C}^k \bigcap \mathcal{C}^j \neq \{\emptyset\}$ then system (8) is feasible. $\mathcal{C}^k \bigcap \mathcal{C}^j \neq \{\emptyset\}$ implies that there exist $\theta^* \in \mathcal{C}^j$ and $\theta^* \in \mathcal{C}^k$ which from (9) and (10) implies that there exist a signal $y^*(t)$ such that $\theta^* = \mathcal{F}_M(y^*(t)), \ y^*(t) \in \mathcal{S}^j$ and $\theta^* = \mathcal{F}_M(y^*(t)), \ y^*(t) \in \mathcal{S}^k$, which means that the signal $y^*(t)$ is both in $\mathcal{S}^j$ and $\mathcal{S}^k$ which implies that it satisfies (8), then (8) is feasible. This completes the proof. □

3. EXPERIMENTS

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. We briefly describe the experiment set up in the following section.

3.1 *Experimental setup*

Our subjects drew using the XPaint program on a PC running Red Hat Linux 7.2 with a screen measuring 1600×1200 pixels and a working window of 700×500 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 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, reported in Figure 2. Each of the 5 subjects was shown the prototypes and was asked to reproduce them on a 700×500 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. Thus we check if reach and draw actions define a well posed pair of movemes according to Definition 2.3.

3.2 *Classification*

We start from the hypothesis that "draws", which are straight lines traced with a specific intention (like drawing a side of the house), and "reaches", which happen with the intention of shifting fast the equilibrium position, define a well-posed pair of movemes. We segmented out by hand a set of straight draws from houses and cars drawn by 2 of the subjects. Reach

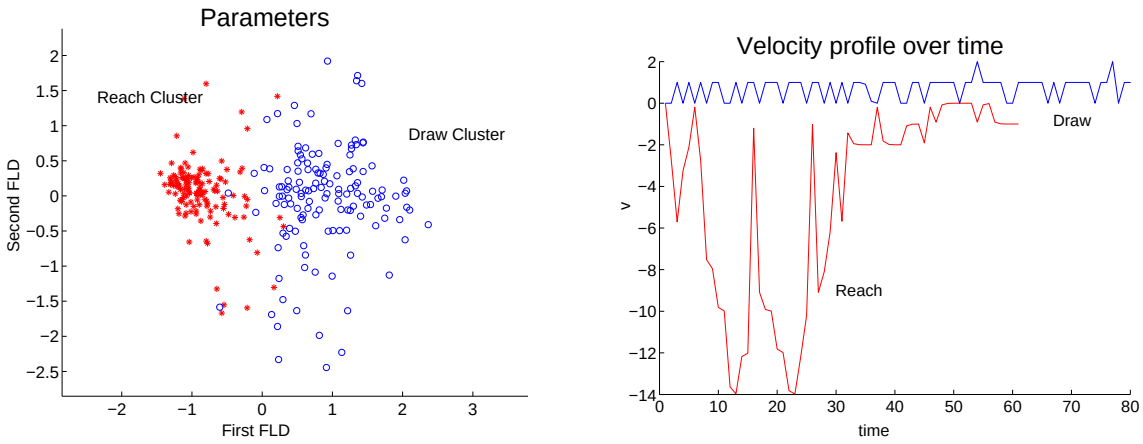

Fig. 3. Parameter estimates for reach and draw examples projected on the first two Fisher linear discriminants (left). Typical velocity profile for reach and draw (right).

examples were obtained from a special experiment session in which the users had to point and click at random buttons appearing on a 700×500 pixels window during a simple video game implemented in MATLAB 6.0.

We considered several dynamical models for representing the reach and draw signals in time, starting from a first order, decoupled model for x and y motion,

$$\dot{x} = a_{1x}x + b_x$$
$$\dot{y} = a_{1y}y + b_y,$$

and proceeding to a second order coupled model,

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ a_{1x} & a_{2x} & a_{3x} & a_{4x} \\ 0 & 0 & 0 & 1 \\ a_{3y} & a_{4y} & a_{1y} & a_{2y} \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ b_x \\ 0 \\ b_y \end{pmatrix}. \tag{11}$$

For estimating the dynamical parameters of all the models proposed we considered their discrete time version so that x, y, $\dot{x}$, $\dot{y}$, $\ddot{x}$, $\ddot{y}$, become x_k, y_k, x_{k+1}, y_{k+1}, x_{k+2}, y_{k+2}. The reach dynamical parameters were estimated from 140 examples of reach trajectories obtained from the video game implemented in MATLAB, and the draw dynamical parameters were estimated from 140 examples of draw trajectories segmented out from cars and houses of 2 of the subjects. The dynamical parameters were estimated for each one of the dynamical models proposed (first order for x and y, decoupled; first order for x and y, coupled; second order for x and y, decoupled; second order for x and y, coupled).

By proceeding with standard pattern recognition techniques (see (Bishop, 1995) for example), we trained a Gaussian classifier for the parameters derived from the 140 examples per class (training set) for each one of the model classes proposed, and obtained the best results for the second order for x and y, decoupled, dynamical model (obtained by letting $a_{3x} = 0$, $a_{4x} = 0$, $a_{3y} = 0$, $a_{4y} = 0$ in system (11)):

$$\begin{pmatrix} \dot{x} \\ \ddot{x} \\ \dot{y} \\ \ddot{y} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ a_{1x} & a_{2x} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & a_{1y} & a_{2y} \end{pmatrix} \begin{pmatrix} x \\ \dot{x} \\ y \\ \dot{y} \end{pmatrix} + \begin{pmatrix} 0 \\ b_x \\ 0 \\ b_y \end{pmatrix}. \tag{12}$$

For such a model we obtained 3.2% training error, and we tested the generalization properties of the resulting classifier on a test set of 323 additional reach examples (obtained from the MATLAB videogame) and 118 additional draw examples obtained from the drawings of other two subjects (different from the ones used for the training set) and obtained 3.63% test error. Figure 3 represents the projection of the parameters belonging to the training set (living in $\mathbb{R}^4$) on the first two Fisher linear discriminants (Bishop, 1995) and typical velocity profiles for the draw and reach trajectories. We let $\hat{\mathcal{C}}^R$ and $\hat{\mathcal{C}}^D$ denote the reach and draw clusters, respectively, according to the notation used in Section 2.3. From the right figure of Figure 3 we notice that a reach trajectory is usually characterized by a bell shaped velocity profile with high velocity variation in a small time, while a draw trajectory is characterized by an almost constant or slowly varying velocity.

3.3 *Well-posedness*

By looking at the sets $\hat{\mathcal{C}}^R$ and $\hat{\mathcal{C}}^D$ of Figure 3 one notice that we have a quite good separation. for the reach motions, so that the system Anyway, since these two sets are just estimates of the real ones $\mathcal{C}^R$ and $\mathcal{C}^D$, we have to check that situation depicted in Figure 1 does not happen. To check this, we find candidate constraints which can describe reach and draw trajectories, so that we may apply Lemma 2.3. Reach trajectories are asymptotically stable with bell-shaped velocity profiles. Draw trajectories are characterized by asymptotic stability properties and by straight lines in (x, y) plane. These requirements for the model (12) imply $a_{1x} = a_{1y}$ and $a_{2x} = a_{2y}$. Some of these parameters are reported in Figure 4 where we can see that their classification is ambiguous since they lie in

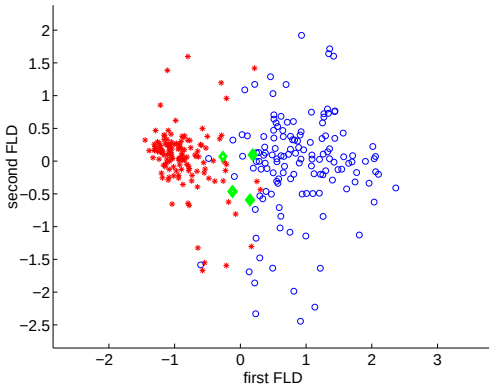

Fig. 4. Diamonds represent some of the parameters corresponding to straight (x, y) trajectories.

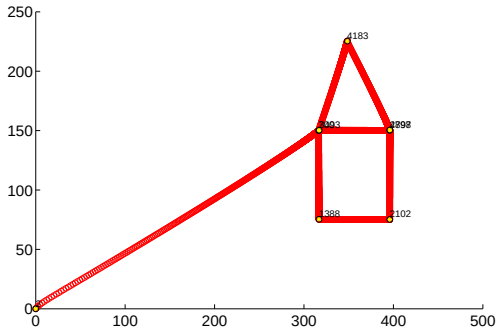

Fig. 5. House generated by points in the overlapping region of clusters of Figure 4.

the boundary region between the first and the second cluster. Then we have a situation analogous to the one reported in Figure 1, where the light sets are $\hat{\mathcal{C}}^D$ and $\hat{\mathcal{C}}^R$ and the dark set is made up by elements like the diamonds in Figure 4. Thus there exist parameters that generate trajectories satisfying both draw (asymptotic stability and straight lines in (x, y) plane) and reach constraints (asymptotic stability and bell shaped velocity profile with high acceleration peak) whose class is undecidable. As an extreme example of this, we show in Figure 5 the shape of a house that has been artificially generated by parameters lying in the region in between the clusters of Figure 4 (dark set of Figure 1), which the classifier classifies as reaches. This happens because the dynamical parameters associated to draw trajectories can significantly differ from each other according to the particular task, and also the velocity profile can consistently vary with respect to the one shown in Figure 3. We show these differences in Figure 6, where we report the draw parameters when a user draws straight lines between two points (as it happens in the draws of the house, ship, car), or a line trying to trace an already existing line, or just a line with no constraints (as it happens in the rays of the suns). We decide therefore to use three classes instead of one for the draw: we call them targeting, tracing, and free motion respectively. Using these definitions, we see from Figure 6 that there is an evident overlapping of the parameter sets of the reach class and free motion class. Therefore we exclude from the panorama the free motion class, and show that the draw class, seen as union of the tracing and targeting motions, can be described in terms of constraints g_D, h_D, g_R, h_R as introduced in Lemma 2.3, such that the system of equations (8) is infeasible. Driven by the characteristics of the velocity profiles of the targeting and tracing draw and reach reported in the bottom right plot of Figure 6, we define the following constraints. The reach trajectories achieve the desired value in a time smaller than a fixed one with respect to a unit step input (which implies a certain acceleration peak), and in the draw trajectories the velocity variation has to be smaller than a given value. We then rewrite these constraints in the form of Lemma 2.3 as $\dot{x} - a\dot{y} = g_D(\dot{x}, \dot{y}) = 0$ and $b - \|(\ddot{x}, \ddot{y})\| = h_D(\ddot{x}, \ddot{y}) \geq 0$ for the draw motions, and $\|(\ddot{x}, \ddot{y})\| - c = h_R(\ddot{x}, \ddot{y}) \geq 0$

$$\begin{aligned} g_D(\dot{x}, \dot{y}) &= 0 \\ h_D(\ddot{x}, \ddot{y}) &\geq 0 \\ h_R(\ddot{x}, \ddot{y}) &\geq 0 \end{aligned} \tag{13}$$

becomes infeasible for suitable b and c. Then if we assume that the constraints above define fair specifications for reach and draw trajectories for the values of b and c that make system (13) infeasible, then the reach and draw classes of signals are well-posed according to Lemma 2.3 . Moreover the $\hat{\mathcal{C}}^R$ and $\hat{\mathcal{C}}^D$ clusters of Figure 3 well represent reach and draw actions, which thus define a pair of movemes M^R and M^D.

4. CONCLUSIONS

We have provided the definition of well-posedness of sets of signals. On the basis of such a definition we provided an operative way to check if sets of actions are well-posed according to a dynamical model class M, and thus they can generate movemes. We have tested our ideas on human drawing data and discovered two sets of actions (reach and draw), which can define movemes and one set of actions (free motion) that is not well-posed according to the already formed alphabet of reach and draw movemes.

5. ACKNOWLEDGEMENTS

This project has been funded in part by the NSF Engineering Research Center for Neuromorphic Systems Engineering (CNSE) at Caltech (NSF9402726).

6. REFERENCES

Bishop, C.M. (1995). *Neural Networks for Pattern Recognition*. Clarendon. Oxford.

Bregler, C. and J. Malik (1997). Learning and recognizing human dynamics in video sequences. In: *Proc. IEEE Conference on Computer Vision and Pattern Recognition*. Puerto Rico. pp. 568–674.

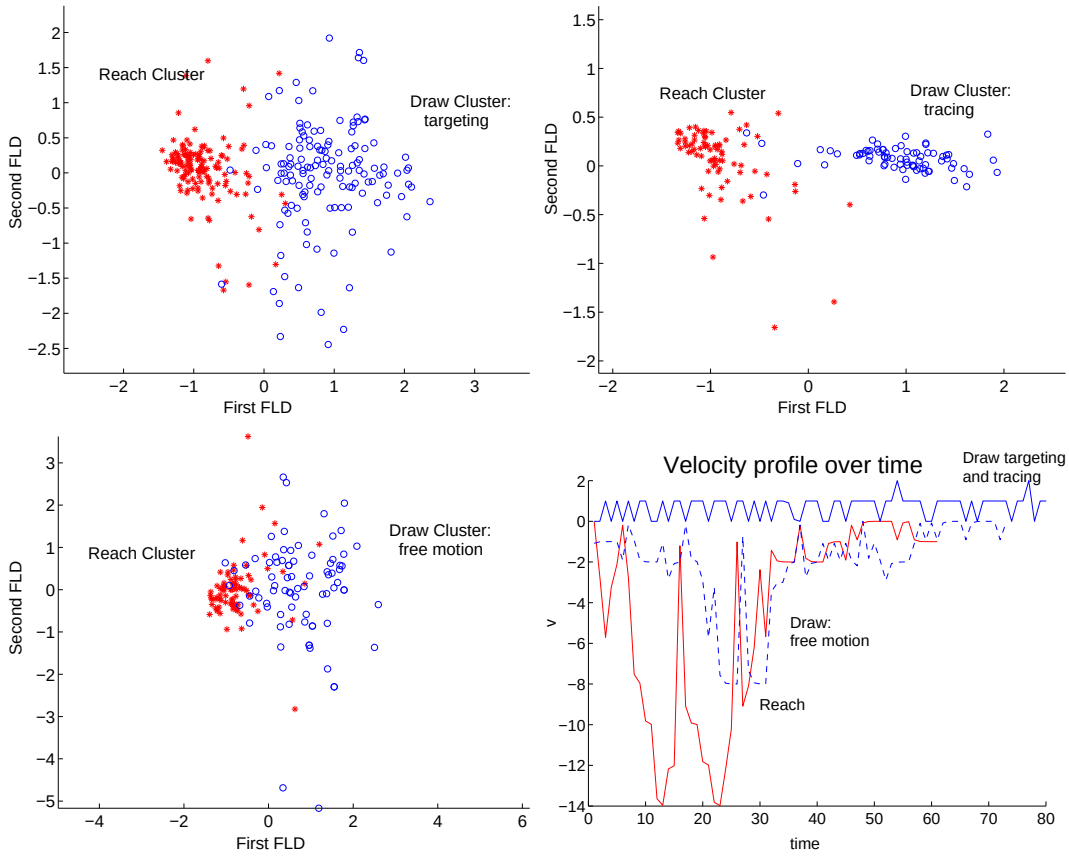

Fig. 6. Parameter estimates for reach and draw targeting, draw tracing and free motion draw with velocity profiles.

Collins, R.T., A. J. Lipton and T. Kanade (2000). Introduction to the special section on video surveillance. *IEEE Trans. on Pattern Analysis and Machine Intelligence* **22**, 745–746.

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

DelVecchio, D., R. M. Murray and P. Perona (n.d.). Segmentation of human motion into dynamic based primitives with application to drawing tasks. In: *Submitted to the 2003 American Control Conference*. Denver, Colorado.

Goncalves, L., E. Di Bernardo and P. Perona (1998). Reach out and touch space (motion learning). In: *Proc. of the Third International Conference on Automatic Face and Gesture Recognition*. Nara, Japan. pp. 234–239.

Goncalves, L., E. Di Bernardo, E. Ursella and P. Perona (1995). Monocular tracking for human arm in 3d. In: *Proc. of the 7th Int.Conference on Computer Vision, ICCV*. pp. 764–770.

Laptev, I. and T. Lindeberg (2001). Tracking of multi-state hand models using particle filtering and a hierarchy of multi-scale image features. In: *IEEE Workshop on Scale-Space and Morphology*. Vancouver, Canada. pp. 63–74.

Ljung, L. (1999). *System Identification*. Prentice Hall. New Jersey.

Munich, M.E. and P. Perona (1996). Visual input for pen-based computers. In: *Proc. of the 13th Int.Conference on Pattern Recognition, ICPR*.

Pedotti, A., P. Crenna, A. Deat, C. Frigo and J. Massion (1989). Postural synergies in axial movements: short and long term adaptation. *Exp. Brain Res. 74* pp. 3–10.

Söderström, T. and P. Stoica (1989). *System Identification*. Prentice Hall. Hemel Hempstead.

Silva, F., L. Velho, P. Cavalcanti and J. Gomes (1997). A new interface paradigm for motion capture based animation. In: *Proc. of 10th Brazilian Symposium of Computer Graphics and Image Processing*. pp. 49–56.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer Verlag.

Waldherr, S., S. Thurn, R. Romero and D. Margaritis (1998). Template-based recognition of pose and motion gestures on a mobile robot. In: *Proc. of the AAAI 15th National Conference on Artificial Intelligence*. pp. 977–982.

Wellner, P. (1991). The digital desk calculator: Tactile manipulator on a desk top display. In: *Proc. of the ACM Symposium on User Interface and Technology*. Hilton Head. pp. 27–33.

Wilson, A. and A. Bobick (1995). Learning visual behavior for gestures analysis. In: *Proc. of IEEE Symposium on Computer Vision*. Coral Gables, FL. pp. 229–234.

Yacoob, Y. and L. Davis (1996). Recognizing human facial expressions from long image sequences using optical flow. *IEEE Trans. on Pattern Analysis and Machine Intelligence 18(6)* pp. 636–642.

Zordan, V.B. and J.K. Hodgins (1999). Tracking and modifying upper-body human motion data with dynamic simulation. In: *Computer Animation and Simulation*. pp. 13–22.