Journal of Visual Languages and Computing (2001) **12,** 657–687
doi:10.1006/jvlc.2001.0202, available online at http://www.idealibrary.com on IDEAL®

High-level Specification and Animation of Communicative Gestures

Sylvie Gibet*, Thierry Lebourque* and Pierre-Francois Marteau†

**LIMSI-CNRS, BP 133, 91403 Orsay, France, sylvie.gibet@univ-ubs.fr,*
†Laboratoire Valoria, Campus de Tohannic-IUP, Rue Yves Mainguy, 56000 Vannes, France

Received 25 February 2000; revised 29 September 2000; accepted 8 January 2001

This paper describes a complete system for the specification and the generation of visual communicative gestures. A high-level language for the specification of hand–arm communication gestures was developed. This language is based on both a discrete description of space, and a movement decomposition inspired by sign language gestures. Communication gestures are represented by symbolic commands that can be described by qualitative data, and translated in terms of spatio-temporal targets that drive the generation system. This approach is well-suited for a class of generation models controlled by key-points information at the trajectory level. The animation model used in our approach is composed of a set of sensory-motor control loops. Each of these models computes in real-time updated angular coordinates of the articulatory structure from the minimization of the distance between target and current locations. At the same time, psycho-motor laws of biological movement are satisfied. The whole control system is applied to the synthesis of communication and sign language gestures. A synthetic character is animated and some results are presented.

1. Introduction

In recent years, gestures in human communication have been increasingly studied. Unlike other modalities such as speech, gestures are bi-directional, acting directly on the environment and reacting to modifications of this environment. In addition, the gesture modality includes a semiotic function because gestures and body movements are capable of driving messages. Human communicative gestures range from simple actions used to designate, point at, or manipulate objects to more complex gestures involved in natural communication where spontaneous gestures co-occur with speech, but can also convey meanings that would be difficult to convey by speech alone.

With the massive development of human–computer interaction, new systems try to take advantage of the expressive power of gestures. At first, gesture interaction had been reduced to simple command interfaces. More recently, techniques such as capturing body movements, recognizing and interpreting human actions and animating virtual humans, have given rise to a number of virtual reality applications with more natural interfaces. However, automatically animating virtual characters with actions that reflect real human motion is still a great challenge. Skilled animators design effective and compelling human characters from key-images or scripts by fastidiously describing every

1045-926X/01/060657+31 $35.00/0

subtle variation of motion. However, to develop more human-like behaviors and real-time control of the virtual characters, we cannot afford time-consuming methods to synthesize motion. Moreover, human movement cannot simply be captured and re-played, but must be parameterized and re-used in various environments. This requires finding the most significant and minimal features that characterize gestures in order to efficiently re-synthesize them in various contexts.

In this paper, we present a system to automatically generate the composing features of hand–arm gesture using a visual gestural language. We aim to avoid both the use of systematically pre-recorded gestures and the specification of tedious sequences of joint movement. The intent of this system is to provide a framework to easily create a wide variety of voluntary movements with economic representation and good interaction time. This system offers the animator a way to manipulate meaningful control parameters that take into account not only communicative goals, but also human style and expression. Moreover, elementary movements built previously or extracted from recorded data could be concatenated or assembled to construct more complex movements. Therefore, the generated gestures should be perceived as natural and should follow biomechanical or psychomotor laws characterizing human movement.

The high-level specification and animation system developed in this research has focused on sign language gestures. These gestures are highly structured and constitute a full-fledged language with a syntax and semantics. Sign languages are the primary mode of communication for many deaf people. Giving the computer the ability to record and to visually represent these gestures would make the interaction of deaf people with their surroundings easier. Furthermore, sign language gestures describe a wide class of upper-limb voluntary movements and can be easily extended to co-verbal gestures, i.e. gestures accompanying speech, or other expressive gestures.

The paper is organized into six sections. Following this introduction, Section 2 highlights related work in the field of movement notation and computer animation. Section 3 presents the architecture of the gesture specification and animation system. Section 4 proposes a description of gesture sentences through qualitative and quantitative specification languages. Section 5 details the complete system for the control and animation of a virtual character. In Section 6, some results of gesture synthesis are presented.

2. Related Works

Representing, specifying and animating human movement requires multi-disciplinary research and so far there are few such works. We selected two categories of studies that show some similarity to our problem. The first one involves the description and notation[a] of body movement and sign language gestures. The second focuses on motion synthesis applied to virtual human animation.

2.1. Movement Notation and Analysis

The attempt to describe body movement has led to different movement notation systems. One system proposed by Rudolph Laban [1], the Laban movement analysis

[a] Notation in this paper denotes a description language or script, as for example music notation.

(LMA), is based on the observations of dance movements and the movement of athletes, martial artists, and people in everyday situations. LMA has evolved into a notation system used in dance and many other movement-related fields. It has four main components: *body*, *shape*, *space* and *effort*. The *body* component deals with aspects such as the initiation of movement from specific body parts and the coordination of movement from a spatial and temporal point of view. It describes which body parts interact with each other and at which time. The *shape* component describes forms that the body makes and any changes in relation to self or the environment. The *space* component is based on spatial patterns and pathways and indicates spatial occupation relative to the environment. Finally, the *effort* dimension is concerned with qualitative aspects of movement. LMA notation is interesting from a spatial representation point of view and could be adapted to represent communicative gestures.

Most systems that describe or transcribe hand–arm gestures derive from studies of sign language. The linguist W.C. Stokoe first proposed a description of American sign language (ASL) in terms of small units of movement he called *cherems*, and a written transcription system based on the combination of these *cherems* [2, 3]. The original notation consists of a limited number of symbols (representing *cherems*), distributed in three classes, each representing one of the formational 'parameters' of a sign: the location of the sign (*tabula or TAB*), the hand shape (*designator or DEZ*) and the movement (*signation or SIG*). A basic assumption of this system is that two signs differ only when one of their distinctive parameters (minimal pairs) changes. In other words, variation in these parameters within a sign is not considered significant. The input of the Stokoe dictionary uses this system to represent ASL signs while respecting the order *TAB, DEZ, SIG.* Movements can be executed in sequence or in parallel. Following Stokoe's work, other studies identified other parameters, such as hand orientation, that play a role in the formation and discrimination of sign [4]. Various notation systems have been proposed, inspired by the original one, which give rise to a family of transcription systems. Other studies are also based on Stokoe's parameters and address the specific problem of sign computing transcriptions [5, 6].

Most of these notations are scripts, composed of symbols or icons that are well adapted to constitute formal databases useful for linguistic studies. However, they are limited in their ability to accurately represent the space around the signer and the dynamics and parallel aspects of gestures.

2.2. Computer Gesture Interaction and Animation

Systems that support hand gestures were developed for interacting with 3D user interfaces. Bordegoni [7] proposed a visual programming environment for designing and recognizing gestures. The design of gesture is done by defining sequences of hand postures and using them by assigning orientation and trajectory values. Gestures are performed by a user wearing a hand-input device and are recognized using pre-specified gestures. In this kind of system, dynamic gestures are reduced to a limited set of hand configurations, the trajectory being specified by a sequence of direction values.

Most virtual human animation systems try to make a virtual person move and act like a real person. Some systems try to animate these virtual characters with meaningful gestures accompanying speech [8–11]. A growing number of other studies are dedicated to the synthesis of sign language gestures to improve communication between deaf and hearing people. These systems integrate techniques from natural language studies and leave the animation of the virtual human itself in the background. Often these systems concatenate pre-stored gestures or use key-framing techniques to compose a sequence of signs. Lee and Kunii [12] implemented a system that translates natural language into sign language using a set of coded hand shapes and pre-defined facial expressions for ASL. Sagawa *et al.* [13] developed a system that translates Japanese sign language into Japanese text and *vice versa*. They use captured hand gestures and replay them to generate sign words. Transitions between words are realized by interpolation functions applied on the end-arm's trajectory and velocity. Losson [14] proposes a complete grammar to specify French sign language (FSL) gestures that are closely linked to linguistic features. The description decomposes signs into four formational parameters (*configuration*, *orientation*, *location and movement*) and follows the Movement-Hold model of Liddell and Johnson [15]. In addition, the movement primitives take into account the symmetrical and repetitive aspects of movement.

Efficient simulation of human behavior can be done with different techniques. Animators frequently use key-frames defined at different times in the animation and compute intermediate frames through interpolation techniques. This technique does not provide the animated characters with actions that look like real human motion unless a huge number of frames is extracted from real captured motions. Therefore, to produce real-time and realistic animations, we cannot afford these time-expensive techniques.

Among real-time methods, the technique of inverse kinematics [16–18] is still widely used in animation. This technique has a low algorithmic complexity and thus low computing cost. However, to enhance the 'naturalness' of the produced movements, this method must include trajectory specifications and pertinent control parameters and weighting functions at the trajectory level.

Other approaches aim to transform or compose movements from elementary pre-defined movements. These techniques involve signal processing algorithms [19–21] to model transitions or integrate user-generated events [22, 23]. The quality of movements produced by such methods depends strongly on pre-defined data. Moreover, they do not provide a straightforward mechanism to vary the expression of the generated movements and do not take into account environmental changes. For higher-level control of generating motions, Zeltzer proposed task-level representation which is based on a hierarchial organization of motor control [24]. A software system, called '*Skill Builder*' combines elementary human motor skills to model complex motion [25]. This is in contrast to other systems based on scripts that are compiled into movement instructions based on a small set of motion primitives [26].

3. Architecture of the Proposed Specification and Animation System

The high-level specification and animation system we seek should:

1. Handle discrete and formal high-level representation of gestures.
2. Provide a way to efficiently describe a large variety of expressive gesture components that can be stored and re-used in different animation contexts.
3. Reduce time between specification and animation of gestures.

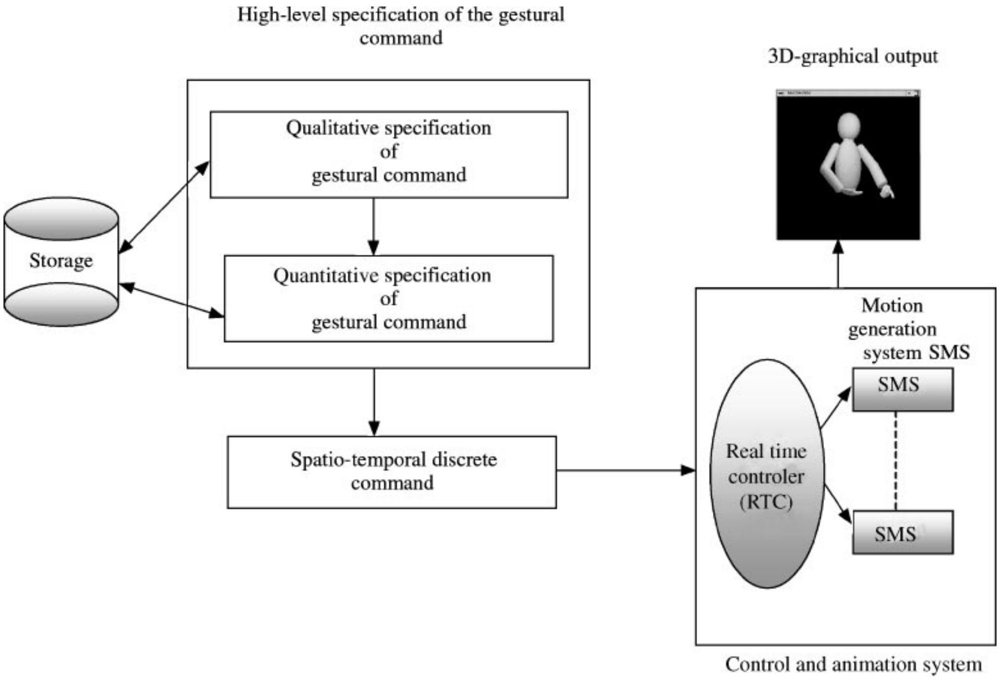


Figure 1. Architecture of the high-level specification and animation system

To achieve these goals, a functional interface should be constructed that ensures adaptation between high-level discrete information representing the gestural command and continuous information that characterizes movement.

The architecture of the proposed system consists of three interacting sub-systems, as presented in Figure 1. A qualitative specification language (QSL) allows the animator to specify qualitative gestural commands with a high-level text-based language that uses no numerical values. These qualitative commands are then translated into quantitative gestural commands that provide spatio-temporal discrete data used to drive the control and animation system. These data are distributed through the real time controller (RTC) to the appropriate animation modules, called sensory-motor systems (SMS). This qualitative specification of gestures allows generic and parameterized gestural commands to be built by means of formal languages. We will see further that the qualitative specification remains rather independent of the animation system, while the quantitative specification is closely linked to the chosen animation technique.

4. Specification of the Gestural Command

This section proposes a framework for a formal spatio-temporal description of communicative gestures. This description is strongly inspired by one used to describe signs[b]

[b] In this paper, the word 'sign' denotes sign gesture in sign language.

in various sign languages. Indeed, we assume that most communicative gestures are found in sign languages (SL) gestures. Furthermore, recent work demonstrates that SL gestures share common structures with co-verbal gestures [27].

4.1. Analysis of FSL Gestures

Past studies of SL have highlighted convergence, not only in the linguistic features of the SL, but also in the formational and functional parameters characterizing gestures. From these linguistic studies [3, 4, 15, 28], it emerges that FSL gestures are composed of five co-occurring features, usually called 'parameters'.

- The configuration that is the hand shape.
- The orientation that gives the directions pointed by the palm and the metacarpus.
- The movement which is generally a description of the arm's endpoint kinematics.
- The location which is the area where the sign occurs. A same sign may be indeed realized in different parts of space, depending on its meaning.
- The facial expression which has a complementary role in the sentence by giving the mode, for example.

As suggested by most SL studies, these parameters have been identified as the smallest units that constitute the fundamental building blocks of the language. We call these units 'gestems', so as to make the manual aspects of sign more explicit.

Our preliminary study was an analysis of the FSL dictionary [29]. Its main significant features were extracted and the frequency of occurrence noted: 1359 signs were observed and stored in a database. This database was originally built for the development of a sign language recognition system [30], and was modified to include information useful for SL description and generation.

In 80% of the FSL signs, configuration is static, i.e. the fingers do not move when other parameters vary. Furthermore, the observations of the signs suggested possible parameterization of the configuration. In fact, a large number of configurations can be described from a small set of basic configurations. For example, a pointing hand shape can be constructed from a basic fist hand shape with an outstretched index finger. Movements were categorized into the five main primitives listed in Table 1. These results highlight the small number of primitives required to describe FSL signs.

Table 1. Proportions of different movement primitives

Primitive	Proportion (%)	Primitive description
Line	42.9	The hand's trajectory in space is a straight line
Arc	26.1	The hand follows an arc in space
Static	17.3	The arm is motionless during the gesture
Circle	10.9	The hand's trajectory is an ellipse
Complex	2.8	The trajectory is more complex than in the other primitives, or is composed of several primitives (movements in zigzag, waves, spirals, etc.)

The analysis of the orientation showed that two kinds are used in SL. The first is the simplest one and we called it *relative orientation* because hand orientation remains unchanged in the wrist referential. It is the most used orientation in FSL. The other kind is called *absolute orientation*. This orientation remains constant in the absolute referential (the center of mass referential), independent of movements of other parts of the body. We also noticed that explicit movements of the wrist are rare. This study yielded important knowledge about the kind and number of primitives needed to specify FSL gestures. It also appears that a discrete representation of the space surrounding the signer must be defined according to the spatial nature of gestures.

4.2. Qualitative Specification of the Gestural Command

In this section, we propose a qualitative description system of the command that prefigure gestures, not the gestures themselves. These commands provide symbolic information to the animation system in charge of producing movement. The qualitative description should be as accurate and exhaustive as possible, yet at the same time, it should provide a generic and parameterizable representation of the command. Moreover, it should be as independent as possible of the generation system and it should provide the animator with an easy way to specify a new gesture with good interaction time.

To meet these needs, we propose a description of gestures based on formational parameters commonly used in SL communities all over the world. That is, we propose a description involving the location, configuration and orientation of the hand, and endpoint movement of the arm. Note that hand location can be implicitly included in the description of arm movement because the arm begins with an initial hand location and completes its movement with a final hand location. Therefore, hand location will not be explicitly considered in our specification. In the following sections, the words configuration and orientation refer to the hand and movement refers to the arms' endpoint.

Before describing the basic primitives characterizing the formational parameters of gestures, we propose a representation of the space around the signer which will be used by these primitives. In the remainder of this section, keywords for the description language are written in bold, and italic typeface represents a non-terminal token. The alternative operator is |. Statements enclosed in brackets are optional.

4.2.1. Spatial Representation

The description of a particular movement in Cartesian space does not have to be very accurate but it does require defining a finite number of key locations or key areas in the reachable workspace of the person performing the gesture. Existing approaches use the notions of orientation and topology to represent the relative position of objects. For example, see the survey proposed by Hernàndez for qualitative spatial representations [31].

We propose a representation of space based on a quantification of the space around the body quite like that proposed by Hernàndez. This representation is approximately centered on the center of mass of the signer, which is considered a neutral point. A location in space is described by combining direction and distance, both chosen in finite sets of data. The space is cut into three main planes (sagittal, frontal and

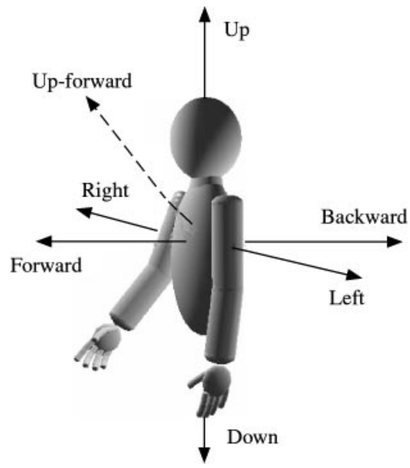


Figure 2. Main directions. Combining the main ones can create new directions

horizontal) and two intermediate planes that form a 45° angle to the frontal plane. Six main directions regarding the planes are then defined: **left**, **forward**, **right**, **backward**, **up and down** (Figure 2). Additional directions can be derived by combining these six. For example, **up** and **forward** generate an intermediate **forward-up** direction at a 45° angle to the horizontal plane. With this system, 17 valid directions can be specified.

In addition, as suggested by Liddell and Johnson [15], we define four magnitudes on each direction: **proximal**, **medial**, **distal**, and **extended**, in order of distance to the body. Finally, the 17 body locations can be used with the primitives, when there exists a contact or proximity notion.

A location in space is specified by one of these expressions:

location ::= **point**(*direction*, [*distance*]) |
point(**near**, *body-location*[,*direction*] |
point(**contact**, *body-location*[,*direction*])

direction ::= **left** | **right** | **forward** | **up** | **down** |
left-forward | **left-up** | **left-down** |
left-forward-up | **left-forward-down** |
right -forward | **right -up** | **right -down** |
right -forward-up | **right -forward-down** |
forward-up | **forward-down**

distance ::= **proximal** | **medial** | **distal** | **extended**

body-location:: = **head** | **ear** | **eye** | **nose** | **mouth** | **chin** | **forehead** | **temple** | **top-head** | **cheek** | **shoulder** | **chest** | **arm** | **elbow** | **forearm** | **wrist** | **hand**

Direction, *distance* and *body-location* have values as just defined. **Point**, **near** and **contact** are elements of the description language. The brackets around the *distance* option mean that this parameter is optional. If it is not specified, **medial** distance is used by default.

When the **near** or **contact** words are used in the **point** expression, direction can indicate **left** or **right** when there is an ambiguity regarding body position.

4.2.2. Movement Primitives

From our analysis of FSL, as described in Section 4.1, we identified seven basic movement primitives. These primitives can be assembled to form more complex movements. *Pointing* movements are unconstrained movements characterized by a location in space reached by the hand. *Straight-line* movements are those where the trajectory of the hand follows a straight line in space. *Curve* movements are those where the trajectory of the hand follows a curved path. *Ellipse* movements draw an ellipse or a circle in space. Among possible complex movements, we retained *wave* movements, characterized by their smoothness, and *zigzag* movements, characterized by overshoot and peaks of acceleration.

In fact, these movement primitives characterize only the trajectory of the arm's endpoint. To reduce the amount of information needed to describe the whole movement and to allow for real-time animation, we retained a class of motion generation systems based on the specification of a limited number of key points. More specifically, movements are described by a set of key positions that prefigure the endpoint trajectory of the arm. As we shall see in Section 5, these positions are not necessarily attained by the arm during the movement itself because of coarticulation and anticipation effects.

The *pointing-mvt* primitive is defined from the specification of one target position. This target can be defined with more or less accuracy and is reached with different velocities. The *straight-line-mvt* primitive uses two and the *curve-mvt* primitive uses three points in space located in the immediate proximity of the trajectory. The *ellipse-mvt* primitive is specified by four points defining the quadrilateral including the ellipse. The *wave-mvt* and the *zigzag-mvt* primitives are described by three points defining one period of the wave (respectively, the zigzag) and a number of waves (respectively, of zigzags). These specifications are illustrated in Figure 3.

The resulting specification is given by the following rules:

movement-primitive :: = *pointing-mvt* | *straight-line-mvt* | *curve-mvt* | *ellipse-mvt* | *wave-mvt* | *zigzag-mvt* | *spiral-mvt*
pointing-mvt :: = **pointing** (*target-location*)
straight-line-mvt :: = **straight-line** (*start-location*, *end-location*)
curve-mvt :: = **curve** (*start-location*, *intermediate-location*, *end-location*)
ellipse-mvt :: = **ellipse** (*start-location*, *first-intermediate-location*, *second-intermediate-location*, *end-location*)
wave-mvt :: = **wave** (*start-location*, *intermediate-location*, *end-location*, *number-of-waves*)
zigzag-mvt :: = **zigzag** (*start-location*, *intermediate-location*, *end-location*, *number-of-zigzags*)

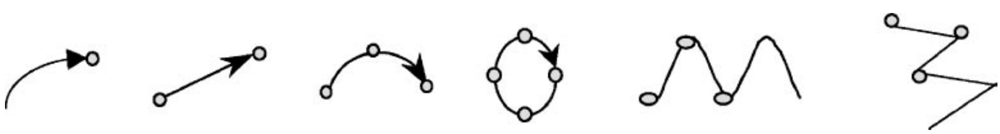


Figure 3. Basic movement primitives: from left to right: pointing, straight-line, curve, ellipse, wave and zigzag

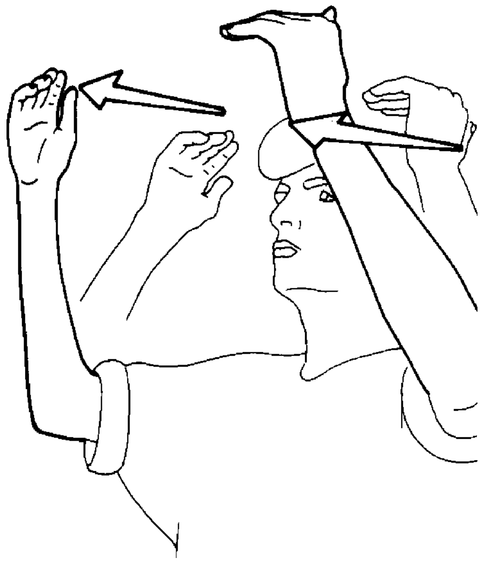


Figure 4. The sign 'ceiling' in FSL

Target-location, *start-location*, *end-location*, *intermediate-location*, *first-intermediate-location* and *second-intermediate-location* are all defined by a *location*. *Number-of-waves and number-of-zigzags* are constant integers. Other complex movements can be derived from these simple ones by combination.

The following figure presents an example of the movement primitive for the sign 'ceiling' in FSL (Figure 4).

The gestural command specification for the right arm is

straight-line (point (right-up), point (right-forward-up))

4.2.3. Hand Configuration

A hand shape can be defined by a set of basic hand configurations with variations on the shape of the fingers. The basic configurations in our specification system are those of the FSL language and they are listed in Figure 5.

A set of modifiers was added so that new hand shapes can be formed from the basic ones by modifying the shape of individual fingers. In our description, fingers are identified by their noun: **thumb**, **index**, **middle**, **ring and little**.

The modifiers are:

- **f-spread, f-clenched** to change the space between fingers,
- **f-angle**, **f-hook** or **f-round** to specify respectively a stretched finger perpendicular to the palm, a hook finger shape and a rounded finger shape.
- **f-contact** to indicate a contact between fingers (essentially between the thumb and other fingers).
- **f-crossing** for the fingers that cross each other (for example, crossing the index and middle finger).

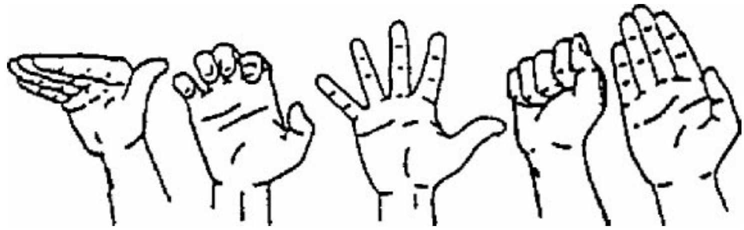
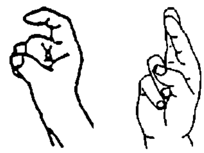


Figure 5. Basic hand configurations: **angle, hook, spread, fist, stretched**



(a) (b)

Figure 6. Configuration **(a)** with rounded index: configuration **(fist, f-round (index)) (b)** the letter "r" expressed by fingerspelling in FSL: configuration **(fist, f-crossing (index, middle), f-foldup (thumb))**

A configuration is then described by the following rules:

hand-configuration ::= **configuration** (*basic-configuration*[, *list-of-modifiers*])
basic-configuration ::= **angle** | **hook** | **spread** | **fist** | **stretched**
list-of-modifiers ::= *modifier* (*list-of-fingers*) | *modifier*(*list-of-fingers*) *list-of-modifiers*
list-of-fingers ::= *finger* | *finger list-of-fingers*
finger ::= **thumb** | **index** | **middle** | **ring** | **little**
modifier ::= **f-spread** | **f-clenched** | **f-angle** | **f-hook** | **f-round** | **f-contact** | **f-crossing** | **f-foldup**

For example, two configurations are shown in Figure 6.

With this description language, most of the configurations used in FSL can be efficiently specified.

4.2.4. Hand Orientation

The hand orientation is described by two directions indicating where the palm and the metacarpus are pointing. The two possible orientations used in SL (relative and absolute orientation) were described in Section 3.1 and are included in our specification language. An orientation is given by

hand-orientation ::= **orientation** (*direction*, *direction* [, *type-orientation*])
type-orientation ::= **relative** | **absolute**

Direction denotes one of the directions of the qualitative spatial description and the optional keyword, *type-orientation*, indicates the type of orientation. The default value is **relative**.

4.2.5. Synchronization and Operators of Gestural Command Modification

Synchronization operators were defined so that starting gestures or gesture components could be specified to occur at particular times. Different relative synchronization operators between gestural command descriptions, given by their identifiers, as well as a repetition operator, are described by the following rules:

synchro-rule ::= *synchro-op1* | *synchro-op2* | *synchro-abs* | *repeat-statement*
synchro-op1 ::= **synchro1 at** (*decal*) **% from** *g-description-id g-description-id*
synchro-op2 ::= **synchro2 at** (*duration*) **{ ms | sec } from** *g-description-id g-description-id*
synchro-abs ::= **synchro-abs after** (*duration*) **{ ms | sec }** *g-description-id*
g-description-id ::= *gesture-id* | *elementary-gesture-id* | *seq-gestem-id*
repeat-statement ::= **repeat** (*num-expr*) *seq-g-descriptions-ids*
seq-g-description-ids ::= *g-description-id* | *g-description-id seq-g-description-ids*
g-description-id ::= **string**
duration ::= *num-exp*
decal ::= *num-expr*
num-expr ::= **float** | **int**

g-description-id is an identifier of the gesture command expression and it specifies either a gesture or a part of a gesture (see Section 4.2.6).

The first operator **synchro1** indicates that the first gestural description begins at τ% after the beginning of the second gestural description. This assumes that the total duration is known or can be estimated. With the second operator **synchro2**, the start of the first gestural description is achieved after a time delay from the beginning of the second gestural description, expressed in seconds. The third operator **synchro-abs** expresses the start of the gestural description after a time delay from the absolute time reference of the simulation. The **repeat** operator allows a sequence of gestural description to be repeated several times.

4.2.6. Gesture Specification

To describe gestures with two arms and two hands moving sequentially or simultaneously, we had to decompose these gestures into sub-gestures executed by different parts of the body. The sub-gestures themselves are decomposed into smaller parts. At the lowest level, we find the *gestems*, which are the formational parameters of the sign, i.e. hand location, configuration and orientation and arm movement. Sequences of individual *gestems* can be combined in parallel to form an elementary gesture. In our application, an elementary gesture is the part of a gesture performed by one upper limb. During a gesture, the elementary gestures occur in parallel or in sequence to form a gesture. Finally, gestures are arranged sequentially to constitute a phrase. Transitions between gestures are included in the specification. We will see in Section 5 how problems usually encountered with these transitions are solved by the generation system.

phrase ::= **begin phrase** *[phrase-id] [synchro-rule] seq-gestures* **end phrase**
seq-gestures ::= *gesture* | *gesture seq-gestures* | *gesture-id [ident-body]*
gesture ::= **begin gesture** *[gesture-id] [synchro-rule] set-of-elementary-gestures* **end gesture**
set-of-elementary-gestures ::= *elementary-gesture* | *elementary-gesture set-of-elementary-gestures*

elementary-gesture	::=	**parbegin elementary-gesture**
		[*elementary-gesture-id*]
		[*ident-body*]
		[*synchro-rule*]
		set-of-seq-gestems
		parend elementary-gesture
set-of-seq-gestems	::=	*seq-gestem* | *seq-gestem set-of-seq-gestems*
seq-gestem	::=	**begin seq-gestem**
		[*seq-gestem-id*] [*ident-body*] *seq-configurations* **end seq-gestem** |
		begin seq-gestem
		[*seq-gestem-id*] [*ident-body*] *seq-movements*
		end seq-gestem |
		begin seq-gestem
		[*seq-gestem-id*] [*ident-body*] *seq-orientations* **end seq-gestem**
ident-body	::=	**left-arm** | **right-arm**
seq-configurations	::=	*hand-configuration* | *hand-configuration seq-configurations*
seq-movements	::=	*movement-primitive* | *movement-primitive seq-movements*
seq-orientations	::=	*hand-orientation* | *hand-orientation seq-orientations*
phrase-id	::=	**string**
gesture-id	::=	**string**
elementary-gesture-id	::=	**string**
seq-gestem-id	::=	**string**

Hand-configuration, *hand-orientation* and *movement-primitives* are descriptions that were presented above. For example, the sign 'crowd' in FSL (Figure 7) is described for the right arm by:

begin seq-gestem
    config-crowd
        configuration(**stretched**, **angle**(**index**))
        configuration(**stretched**, **angle**(**middle**))
        configuration(**stretched**, **angle**(**ring**))
        configuration(**stretched**, **angle**(**little**))   **end seq-gestem**
parbegin elementary-gesture
right-arm
  repeat(**3**)   config- crowd
  begin seq-gestem **straight-line**(**point**(**right**, **medial**), **point**(**left**, **medial**))   **end seq-gestem**
  begin seq-gestem **orientation**(**down**, **left**, **absolute**)   **end seq-gestem**
parend elementary-gesture

Notice, however, that we have only considered synchronization operators that synchronize gesture components at the beginning of gestural descriptions. We might also include operators that allow synchronization at the end of the descriptions. Actually, when different gesture components are executed simultaneously, we implicitly assume that all the resulting processes are waiting for the end of the longest process.

Figure 7. The sign 'crowd' in French Sign Language

4.3. Quantitative Specification of Gestural Command

A gestural qualitative command, expressed as a text-based program, is translated into a set of low-level instructions that produce sequences of data used directly by the animation engine to control the avatar.

The notion of 'target' was introduced when motion primitives were described (Section 4.2). Targets are associated with a motor task and represent a goal to attain or a motor intent prior to execution of motion. They are represented by location in Cartesian space defining the endpoint of an arm's reach. In the next section, we extend this notion of target by adding an accuracy criteria. For this, a target becomes a sphere and the motion attains its goal when it enters the area of the sphere.

Quantitative gestural commands are specified by an imperative language based upon a description of targets and mechanisms to assemble them in space and time. The general structure of this language is given by:

- The specification of individual targets. Spatial targets for movement or configuration targets for hand orientation or hand shape will be distinguished in the next section.
- The definition of sequences of targets corresponding to the primitive parameters of the gesture (hand configuration, hand orientation or movement).
- The definition of subgestures specified as parallel sequences of targets.
- The definition of a gesture that appears to be a composition of elementary gestures. Subgestures are combined in parallel.
- The definition of sequences of gestures that constitute gestural phrases.

Each level of definition can be identified and referenced in other portions of code. Synchronization mechanisms at the target level have also been included in the language. This is achieved by specifying the timing of the target occurrences. The link with the qualitative description of gestures is relatively simple. Different levels of gesture composition previously identified can be directly translated into a quantitative gestural command specification, based on the assembling of spatio-temporal targets.

5. The Control and Animation System

We consider at the generation level a set of sensory-motor systems (SMS), each one being attached to a particular articulated chain that composes the upper-limb system of the virtual human body. We propose 12 SMS to handle the two hand–arm systems. One SMS is associated with each arm and each finger. Each SMS produces an elementary movement. Target locations, representing goals to be reached by the articulatory chain endpoints, are used to control the SMS. These targets or goals prefigure the motion and reflect some kind of motor intent for voluntary movements. It is not necessary to build entire trajectories of the arm's endpoint or use predefined ones. An SMS automatically converts discrete orders into continuous data characterizing the movement. The gestural command level is thus reduced to composition of discrete targets in space and time, directly linked to specification primitives, and fed into the SMS.

5.1. The Motion Generation System

The basis and principle features of the (SMS) were originally presented in Gibet & Marteau [32] and exploited for animation purpose in Gibet & Lebourque [33, 34]. These principles are rapidly recalled. They are then applied to the control of a geometrical hand–arm system with joint constraints.

5.1.1. The Sensory-Motor System (SMS)

The control of articulated chains with large number of degrees of freedom can be considered as an 'inverse problem' because the state control variables of the effectors implied in the movement must be computed from sensory information observed during the execution of the movement. We solve this problem by using a formalism inspired by closed-loop control systems. In a motor task, a set of effectors responsible for the execution of motion is selected, as well as sensory receptors whose data are captured at each time iteration to control the space–time coordination of the effectors. The error signal measuring the distance between the output of the system during performance and a reference value will be used to iteratively update the state of the system.

The sensory-motor system involves three working spaces (Figure 8):

- A state space where knowledge of state vector $\mathbf{q}$ in time completely defines the motor system.
- An observation space in which sensory data $\mathbf{a}$ can be observed from specific sensors.
- A task space in which the specification of tasks can describe a motor program. Let $\mathbf{a}_t$ be the task vector.

The inverse problem can be stated as the problem of finding the appropriate state vector $\mathbf{q}$ from the specification of a current vector $\mathbf{a}$ in the observation space and a task vector $\mathbf{a}_t$ in the task space. M is the direct transformation which links the observation vector $\mathbf{a}$ to the state vector $\mathbf{q}$: $\mathbf{a} = M(\mathbf{q})$. If we consider a scalar cost function E, measuring the error magnitude between the task vector $\mathbf{a}_t$ and the observation vector $\mathbf{a}$, the inversion of direct transformation can be regarded as iteratively updating the state vector $\mathbf{q}$ to minimize the cost function E.

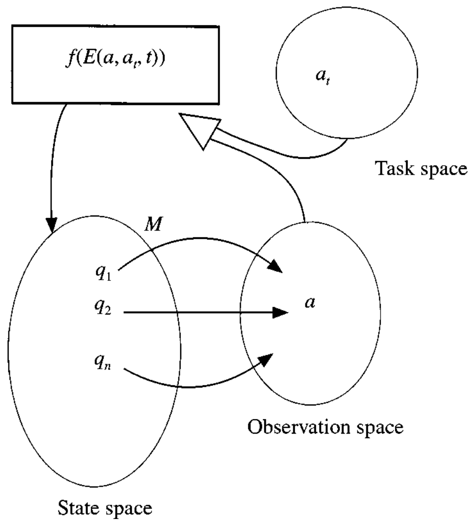


Figure 8. Three working spaces

The solution to this problem is given by the following set of differential equations:

$$\frac{\partial \mathbf{q}}{\partial t} = -g(E(\mathbf{a}, \mathbf{a}_t, t)) \cdot grad(E(\mathbf{a}, \mathbf{a}_t, t))$$

$$= -g(E(\mathbf{a}, \mathbf{a}_t, t)) \cdot \left(\frac{\partial M}{\partial \mathbf{q}}\right)(M(\mathbf{q}) - \mathbf{a}_t) \tag{1}$$

$(\partial M/\partial \mathbf{q})$ is the Jacobian matrix of the operator M, g is a gain function and *grad* the gradient operator.

When applying this inversion principle to the control of an articulated geometrical chain (Figure 9), $\mathbf{q}$ represent the angular state vector, $\mathbf{a}$ the endpoint location of the chain and $\mathbf{a}_t$ the target location. For a model of the human arm, an elementary task consists of assigning a desired position to the arm's endpoint. The state vector is automatically updated according to the gradient descent of the error magnitude calculated between the current observed location $\mathbf{a}$ and the spatial target to reach $\mathbf{a}_t$.

Note that if we use direct feedback with a constant gain function, we will not be able to damp large angular variations calculated through the error E. In particular, transitions on $\mathbf{q}$ might trigger instabilities in some configurations. To ensure the stability of the system and to generate damped behaviors, a nonlinear function and a second-order filter have been introduced in the control function, as shown in Figure 10.

The nonlinear function has a 'sigmoïd' shape. The gain of this function increases exponentially when the error between the observable position and the reference target position is reduced. The stability and asymptotic properties of this model was proved in Gibet & Marteau [32].

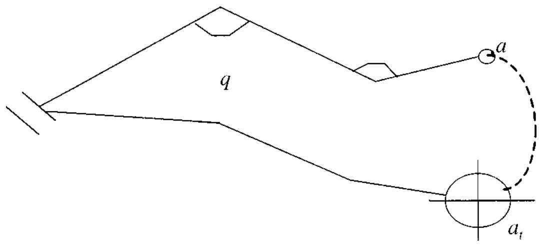


Figure 9. Representation of an articulated chain associated to a spatial target, where a is the observation vector giving the position of the chain end-point, a_t is a spatial target

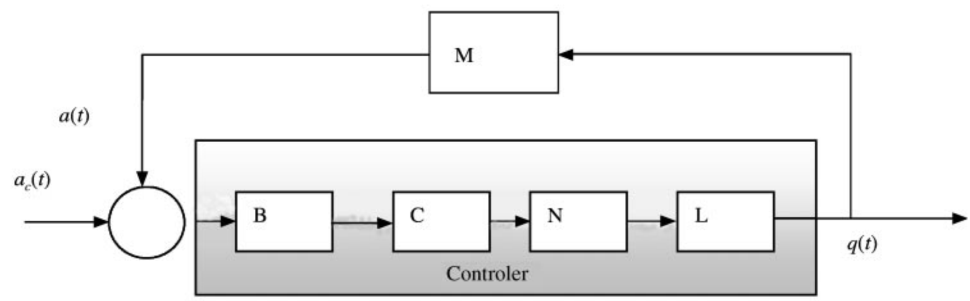


Figure 10. Sensory-motor control loop: $B(a(t), a_t(t)) = -grad\{E(a(t), a_t(t))\}$, where E is the quadratic error between a and a_t, C is a matrix that models the articulatory constraints, N is a non-linear function, L is composed of a second-order filter and an integrator, and M is a production operator: $a = M(q)$

5.1.2. Coarticulation

More complex gestures can be synthesized by specifying a sequence of targets in Cartesian space at the command level. However, this technique does not guarantee smoothness of the produced movements, or the anticipation of movement that consists of adapting movement according to the context of execution. Therefore, at each iteration step, variation in the angular coordinates of the effector system should depend on previous movements as well as on incoming movements. One way to model this co-articulation effect is to take into account the influence of targets from the near past and the near future (Figure 11). This can be achieved if the error function E is modified so that it becomes a weighted sum of elementary costs, each elementary cost being activated during a time segment centered on each target occurrence in the sequence:

$$grad(E(\mathbf{a}, \mathbf{a}_t, t)) = \sum_{j=i-1}^{i+1} \lambda_j(t) \left(\frac{\partial M}{\partial \mathbf{q}}\right)(\mathbf{a}_j - \mathbf{a}_{t_j}) \text{ with } \lambda_j(t) = K\mathrm{e}^{-(t-t_j)^2/\tau_j} \quad (2)$$

With this modified cost, which contains a product of spatial terms and temporal function, some articulators may be pre-positioned in order to reach the subsequent

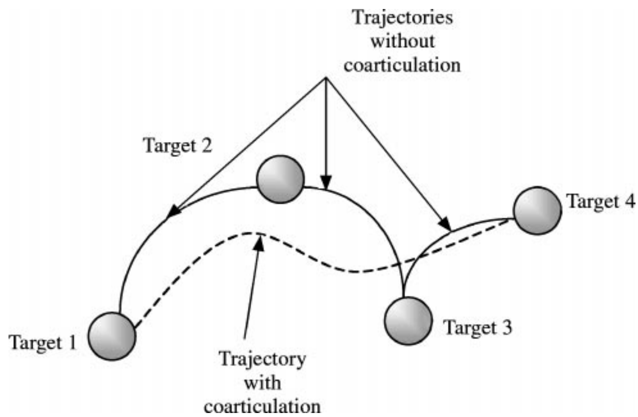


Figure 11. Emergent targets (a) and coarticulation (b)

targets, even if the articulated chain endpoint does not move. The co-articulation effect also allows the concatenation of consecutive movements with smooth transitions.

We suppose that the targets are sufficiently distant so that we only need to consider the targets (Ci_{i-1} and Ci_{i+1}) adjacent to Ci_i.

5.1.3. Geometrical Models of the Hand–Arm System

An articulated chain models the arm. It is composed of three segments representing the arm, the forearm and the hand. It has seven degrees of freedom (Figure 12), and joint constraints approximated from anthropometric data are given in Table 2.

The hand is modeled by a set of five articulated chains, each one having four degrees of freedom (Figure 13). Joint constraints are also given in Table 3. They are identical for each finger, except the thumb. The fingers are modeled by a simplified closed-loop system, controlled by targets expressed in angular space (configuration targets).

Note that the different degrees of freedom of the fingers are not completely independent. There is a linear relation between the angles of the two last articulations of the fingers:

$\theta_3 = 2\theta_2/3$, with θ_2 and θ_3 being the second and the third angle of the phalanxes [35].

5.2. The Real-Time Controller (RTC)

The arm is decomposed into seven articulated chains: one for the arm itself (from the shoulder to the wrist), one for the wrist and five for the hand. Therefore, seven motion generators running in parallel are used to control one hand–arm system.

The control of the arm is achieved by an SMS as described in Section 5.1. The link with the quantitative gestural command is achieved by specifying a sequence of spatial targets X_T^i occurring at time t_i: $\{X_T^i, t_i\}$. These targets can be co-articulated or not, depending on the desired smoothness of the motion.

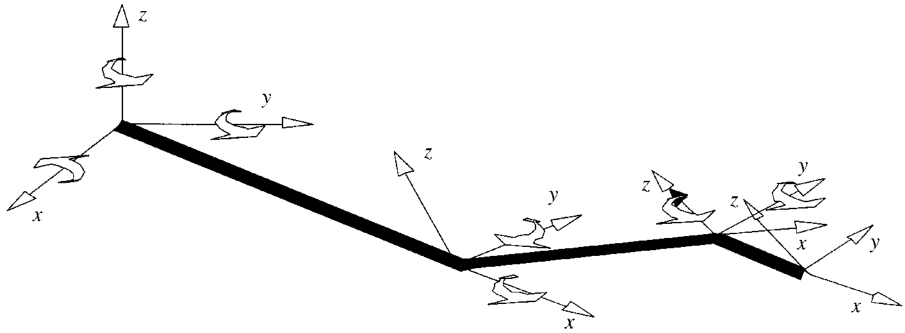


Figure 12. Geometrical model of the arm

Table 2. Joint constraints of the arm

	x		y		z	
Axis of rotation	Min (deg)	Max (deg)	Min (deg)	Max (deg)	Min (deg)	Max (deg)
Shoulder	− 10	110	− 100	90	− 10	180
Elbow	− 180	90	0	170	—	—
Wrist	—	—	− 20	90	− 30	20

Each target X_T^i is defined as a location in Cartesian space associated to an accuracy parameter defining a sphere. A target is reached when the articulated chain endpoint enters the accuracy sphere associated with the target. This allows the control system to parameterize accuracy at the end of the motion and velocity endpoint.

The control of the hand is relatively simple. It is fastidious indeed to control each articulated chain representing the fingers by targets located at the fingers' endpoint trajectories. Therefore, a global hand configuration at the end of the movement is defined. A 'configuration target' is the angular vector (in state space) representing the angular coordinates of the fingers. For configuration targets, we define also an accuracy parameter that is the maximum angular variation allowed for the final configuration vector. Note that this control is a direct control as there is no inversion of the Jacobian matrix.

In conclusion, the quantitative gestural command of the system is reduced to a sequence of spatio-temporal targets. They are either configuration or spatial targets and they are dedicated to one of the articulated chains of the hand–arm system. These targets are specified directly from the quantitative description given in Section 4.3 or they are calculated automatically from a qualitative specification of the gestural command (Section 4.2).

Before the SMS can pilot the complete arm, the controller receives all the targets resulting from a given specification (Figure 14) and assigns them to appropriate loops at the proper time. This controller is cadenced by an external clock that provides a time reference needed to coordinate the whole system.

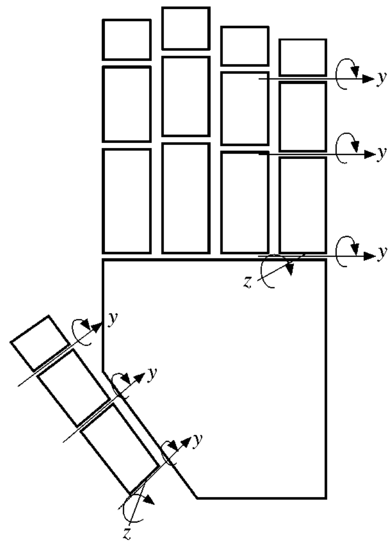


Figure 13. Geometrical model of the hand

Table 3. Joint constraints of the fingers

	Thumb				Other fingers			
	y		z		y		z	
Axis	Min (deg)	Max (deg)	Min (deg)	Max (deg)	Min (deg)	Max (deg)	Min (deg)	max (deg)
First phalanxes	0	45	60	0	0	90	30	15
Second phalanxes	0	90	—	—	0	90	—	—
Third phalanxes	0	90	—	—	0	90	—	—

The controller also verifies that the movement is correctly performed and that there are no problems during execution.

6. Animation Results

The results presented in this section illustrate the merits of the animation method for generating natural dynamic gestures. Two evaluation methods were performed: a comparison of synthesized signals with real ones, and a verification of invariant laws characterizing human movement. Furthermore, this section highlights the ability of the high-level qualitative language to create new gestures from scratch or from the composition of previously designed gestural elements. We insist that information at the gestural

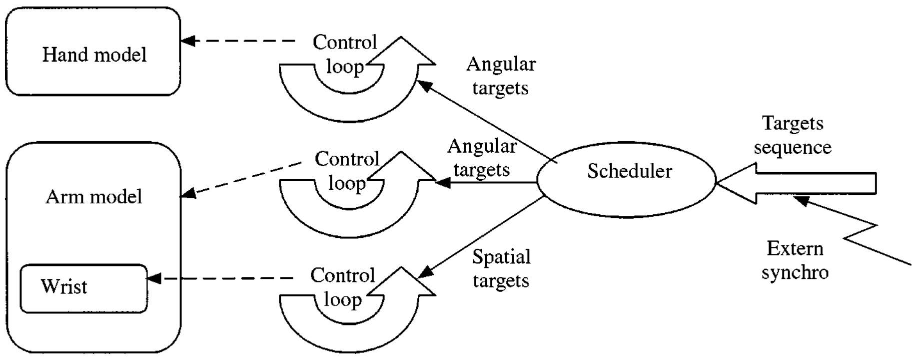


Figure 14. Control of the hand–arm system

command level be compact and that there be good interaction time between new specifications and the production of animated sequences.

6.1. The Animation System

The animation system was implemented on SGI O2, and Pentium Linux architectures. The 3D models are achieved with Open GL (Figure 15). The animation software runs in real time with a simulation rate of 100 points per second for computing the angular coordinates. Graphical 3D output is generated at a rate of 8 frames per second.

6.2. Comparison of Real and Synthesized Dynamic Gestures

To compare synthetic and real gestures, a corpus of gestures was built from recordings of primitives in the qualitative specification system. Two different subjects performed each primitive two times with three amplitudes. The recordings were made with a Flock of Birds from Ascension Technology. Electromagnetic sensors that measure a position and an orientation were used to record the position of the hand during a gesture with an acquisition frequency of 40 Hz. Data were filtered and speed, acceleration and radius of curvature were computed for each gesture. Synthetic data from the animation engine was compared to real data primarily via velocity profiles. Human curved gestures are characterized by a double-bell-curved speed profile. The animation engine reproduces this feature as shown in Figure 16.

The curves presented in Figure 17 show the trajectories of a wave gesture captured from real gestures and the corresponding synthesized wave gesture trajectories.

6.3. Invariant Laws Characterizing Biological Movements

Signals captured from real movements verify experimental laws that characterize variability in human movement under specific conditions. This variability has been explored by biomechanists, especially in the field of motor control and skill acquisition. From these studies, general laws underlying the spatio-temporal organization of motricity have been highlighted.

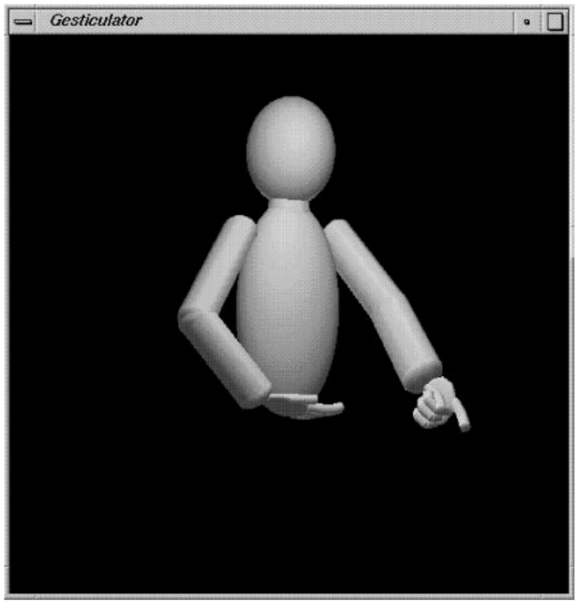


Figure 15. 3D-graphical output window

In particular, for pointing movements, Fitt's law [36] expresses a logarithmic relation between movement duration, accuracy and movement time:

$$T = a + b \log \frac{2D}{W} \tag{3}$$

T is the movement duration, D the movement distance, W the target width (providing the notion of accuracy), and a and b are experimentally determined constants.

Simulations of the geometrical arm were performed. Our SMS verifies Fitt's law for simple pointing movements, as can be seen in Figure 18.

For curved repetitive movements, we used results from Viviani [37], comparing both speed profiles and the ratio $V/R^{1/3}$, where V is the tangential velocity and R the radius of curvature. This ratio was computed from normalized data. Curves are shown in Figure 19.

6.4. Generation of Signs from the Specification of Gestural Commands

Finally, we produced signs and small sentences in FSL. The qualitative language was used to specify these signs and sentences. The description files are very compact. For example, only 60 lines were needed to describe a sentence consisting of four signs.

The first example illustrates the synchronization process, involving the two hand–arm systems (Figure 20).

The high-level specifications of the signs *Thank you*, *You*, *Listen to* and *Me* for the sentence 'Thank you for listening to me' are expressed below. The sign *Thank you* is

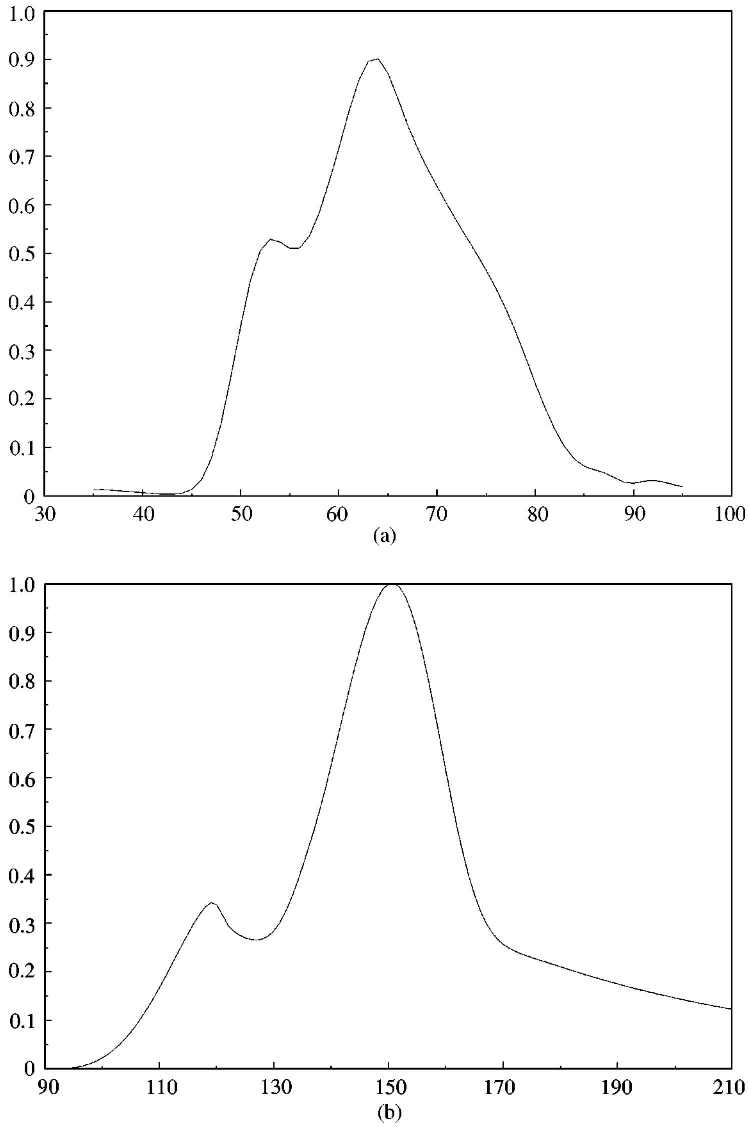


Figure 16. Normalized speed profiles for curved movements. Real gesture is on the upper curve; synthesized gesture is on the bottom

shown in Figure 21. The animation result is shown in Figure 22.

```
begin gesture Thank you
    parbegin elementary-gesture
        begin seq-gestem
            point(near,chin)
        end seq-gestem
        begin seq-gestem
```

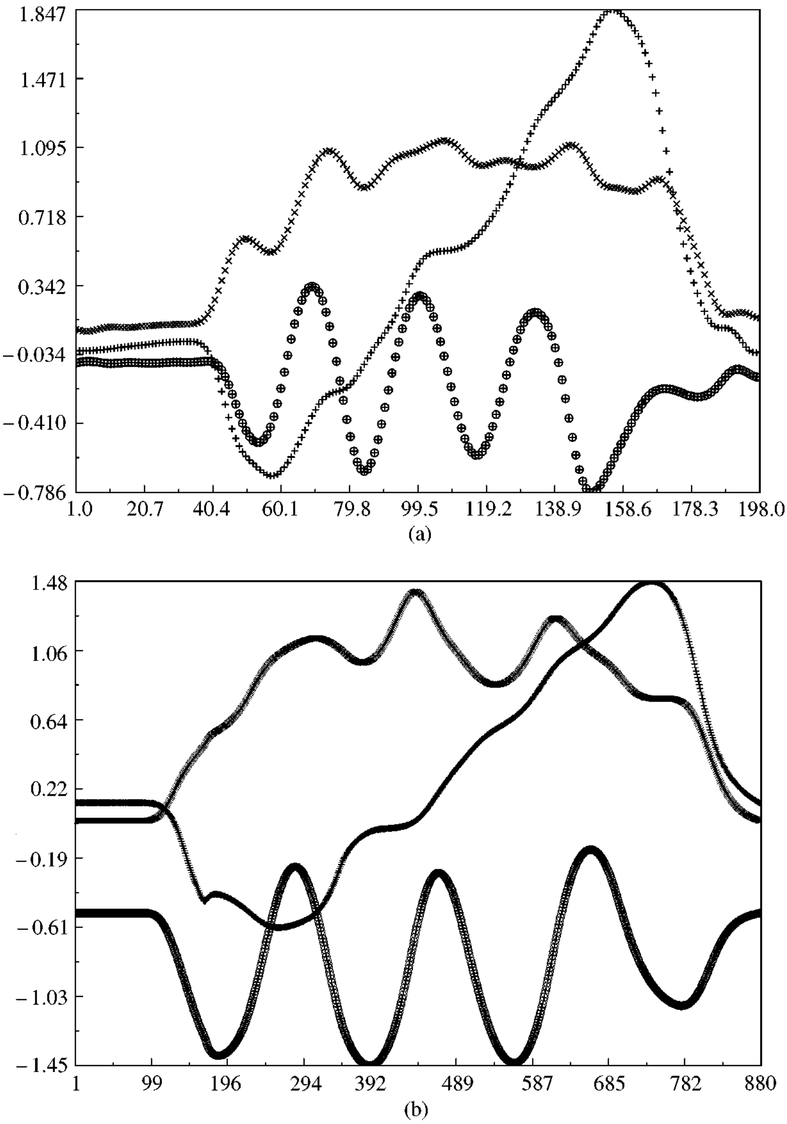


Figure 17. x, y and z positions for wave movements. Real gesture is on the upper curve; synthesized gesture is on the bottom: $+$, x; $\times$, y; $\oplus$, z

```
        configuration(stretched)
end seq-gestem
begin seq-gestem
        orientation(up, forward)
end seq-gestem
```

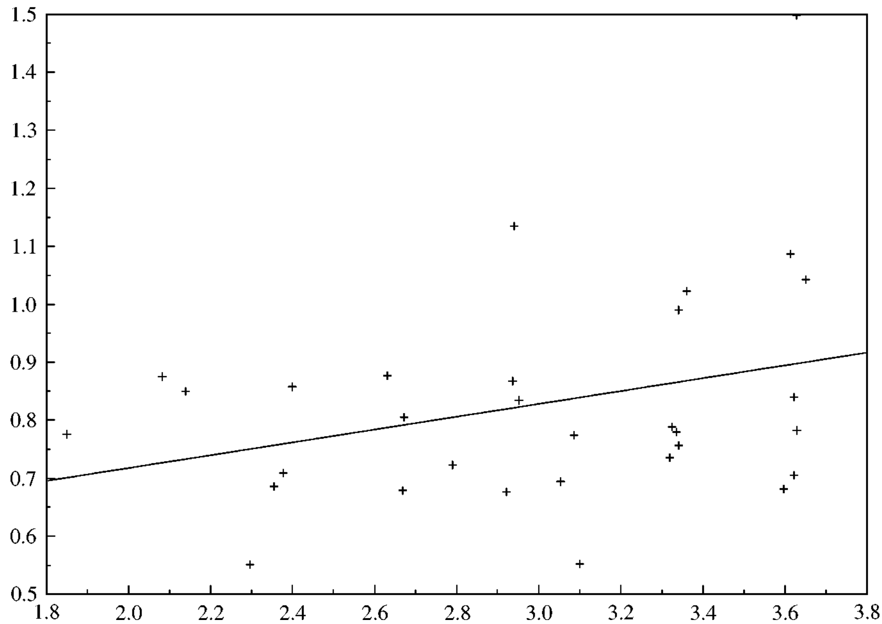


Figure 18. Fitt's law. The abscissa corresponds to the calculus of $\log(2D/W)$, the ordinate to the duration of the pointing. Data coming from the model are represented by the crosses. The line has been calculated from these data by linear regression: $+$, Donnees collectees: abscisse: $\log(2D/W)$, ordonnee: T; ——, $y = 0.4960255 + 0.1106464x$

```
        parend elementary-gesture
        parbegin elementary-gesture
                begin seq-gestem
                        point(forward, medial)
                end seq-gestem
                begin seq-gestem
                        configuration(stretched)
                end seq-gestem
                begin seq-gestem
                        orientation(up, forward)
                end seq-gestem
        parend elementary-gesture
end gesture

begin gesture You
        parbegin elementary-gesture
                begin seq-gestem
                        point(forward, distal)
                end seq-gestem
                begin seq-gestem
                        configuration(fist, f-spread(index))
                end seq-gestem
```

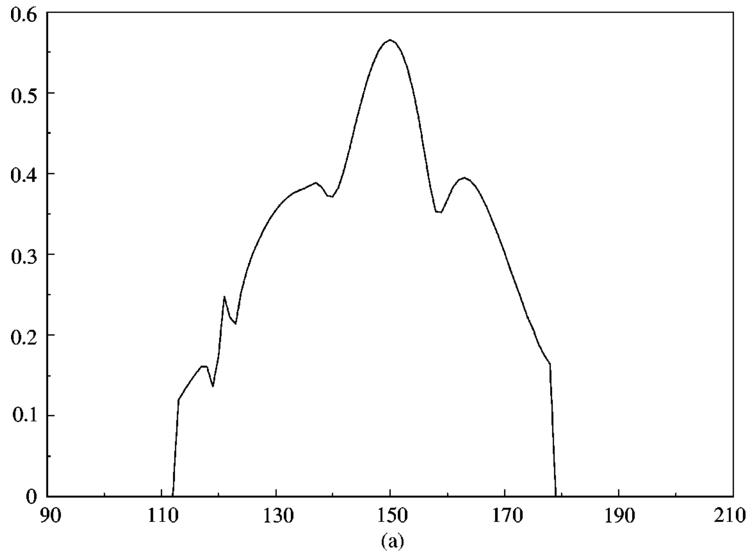


Figure 19. Two-third law for curved movements. Synthetic data are on the upper curve

```
        parend elementary-gesture
end gesture

begin gesture Listen to
        parbegin elementary-gesture
                begin seq-gestem
                        point(contact, ear, right)
                end seq-gestem
                begin seq-gestem
```

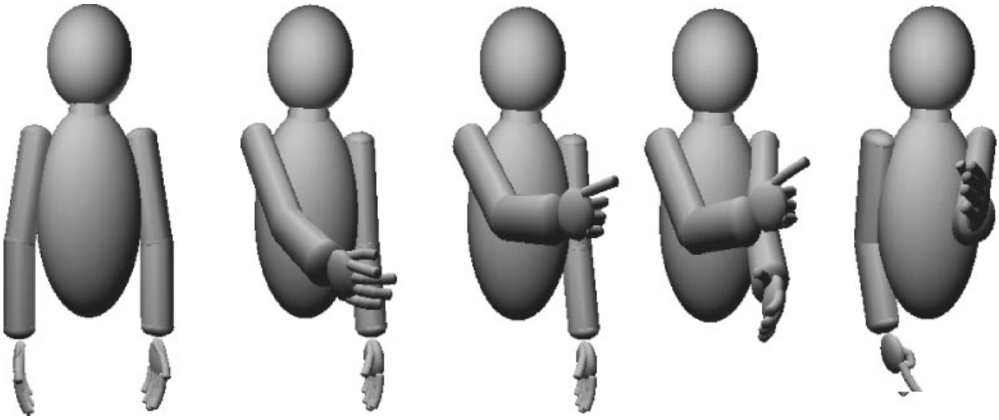


Figure 20. Example of gesture with synchronization

Figure 21. Specification of the sign 'thank you' in FSL

Figure 22. Animation corresponding to the sentence 'Thank you for listening to me' in FSL

```
                configuration(fist, f-spread(index, thumb, middle))
            end seq-gestem
            begin seq-gestem
                orientation(up, forward)
            end seq-gestem
        parend elementary-gesture
end gesture
```

```
begin gesture Me
    parbegin elementary-gesture
        begin seq-gestem
            point(contact, chess)
        end seq-gestem
        begin seq-gestem
            configuration(fist, f-spread(index))
        end seq-gestem
    parend elementary-gesture
end gesture

begin phrase Thanks
    Thank you right-arm
    You right-arm
    Listen to right-arm
    Me right-arm
end phrase
```

7. Conclusion and Perspectives

A system for high-level specification and generation of communicative gesture was applied to the control and animation of an anthropomorphic model of two hand–arm systems. In this system, a gestural command is specified qualitatively and then gets translated into a numerical specification through an imperative language. First, the space around the signer is discretized by defining specific directions. Qualitative attributes for distance then define a location in space and a contact attribute for the body. Arm movements are categorized according to their main families: pointing, straight lines, curves, circles and waves. Hand movements are discriminated according to relative and absolute orientation and a finite set of configurations is defined for the hands and fingers. These basic components and primitives, which are similar to those identified for sign language gestures, are assembled in sequence or in parallel to form complex gestures. This study on hand–arm movements provides the first lexical and syntactic elements for specifying gestures. In addition, time synchronization operators further describe elementary movements.

One important aspect of our system is that the specification and control system is independent of the generation system. This last system is composed of a set of sensory-motor loops associated with each articulated chain representing the arms or the fingers of the hand. Each sensory-motor system has the ability to automatically generate natural trajectories of movement from the specification of targets in a visual representation space. Furthermore, the generated movements reproduce the main psychomotor invariant features of human movement. Therefore, this architecture makes it possible to describe a complex, continuous gesture with a very little information, reduced at the command level to a set of discrete spatio-temporal targets. Because the system takes co-articulation into account, it produces smooth gestures and sequences of elementary movements can be concatenated without specifying transitions. So far, the evaluation of this system has verified some of the invariant rules of human motion and has compared synthesized movements with real ones.

While this system is capable of generating sign language gestures, it can also generate any kind of communicative gesture. Several extensions of this work can be proposed. We plan to extend our specification language to integrate more elaborated syntactic and semantic models of sign languages, in particular, FSL. We have already seen that grammatical attributes are taken into account at the movement generation level. These attributes are represented not only by kinematics and dynamic parameters and location assignment, but also by mechanisms such as repetition, duration and pauses between signs. To build sentences that are not solely based on the concatenation of signs, it is necessary to include these parameters in the qualitative specification language. For example, the order of the signs should depend on the message of the sentence rather than its syntax. Finally, the complete synthesis of sign language gestures necessitates the synthesis of facial mimics and expressions. They can be used alone, for example, to mark the type of the sentence (e.g. interrogative or negative) or can be used jointly with the manual characteristics of signs. We also plan to perceptually evaluate the specification and animation system by examining how well people who use French sign language interpret the gestures. This system will constitute a basis for the construction of a complete translation system from text to sign language.

The generation system has recently been extended to control the whole body to synthesize more complex movements that require coordination between limbs and strategies of motor planning. The applications can be juggling, musical gestures or dancing. Although synchronization mechanisms have been included in the specification language presented above, it is difficult to synchronize complex movements where actions are performed in parallel and depend on the environment. For models of these behaviors, it is more efficient to use parallel and reactive languages where reactions to external actions or events are more naturally taken into account. Ongoing research is devoted to implementing reactive capabilities of synthesized characters by means of hierarchical and interacting reactive modules.

References

1. V. Maletic (1987) *Body, Space, expression: The Development of Rudolph Laban's Movement and Dance Concepts*. Mouton de Gruyte, New York.
2. W. C. Stokoe (1972) *Semiotics and Human Sign Language*. Mouton, The Hague.
3. W. C. Stokoe, D. Casterline & C. Croneberg (1978) *A Dictionary of American Sign Language on Linguistic Principles*, revised edn. Linstok Press, Silver Spring.
4. E. S. Klima & U. Bellugi (1979) *The Signs of Languages*, 2nd edn. Harvard University Press, Cambridge, London.
5. S. Prillwitz, R. Leven, H. Zienert, T. Hanke & J. Henning (1989) *HamNoSys, version 2.0, Hamburg Notation System for Sign Languages—An Introductory Guide.* Signum Press.
6. V. Sutton (1998) The SignWriting Literacy Project. In: *Impact of Deafness on Cognition AERA Conference*, San Diego, CA.
7. M. Bordegoni (1994) An environment for the specification and recognition of dynamic gestures. *Journal of Visual Languages and Computing* **5,** 205–225.
8. J. Cassell, C. Pelachaud, N. Badler, M. Steedman, B. A. Horn, W. Becket, B. Douville, S. Prevost & M. Stone (1994) Animated conversation: rule-based generation of facial

expression, gesture and spoken intonation for multiple conversational agents. *Computer Graphics*, *Annual Conference Series*, *ACM*, pp. 413–420.

9. Becheiraz & D. Thalmann (1998) A behavioral animation system for autonomous actors. *Workshop on Embodied Conversational Characters.*
10. D. T. Chen, S. D. Pieper, S. K. Singh, J. M. Rosen & D. Zeltzer (1993) The virtual sailor: an implementation of interactive human body modeling. *Proceedings IEEE 1993 Virtual Reality Annual International Symposium*, Seattle, WA.
11. T. Calvert (1991) Composition of realistic animation sequences for multiple human figures. In: *Making Them Move: Mechanics, Control, and Animation of Articulated Figures* (N. I. Badler, B. A. Barsky & D. Zeltzer, eds). Morgan-Kaufmann, San Mateo, pp. 35–50.
12. J. Lee & T. Kunii (1993) Computer animated visual translation from natural language to sign language. *Journal of Visualization and Computer Animation* **4,** 63–78.
13. H. Sagawa, M. Ohki, T. Sakiyama, E. Oohira, H. Ikeda & H. Fujisawa (1996) Pattern recognition and synthesis for a sign language translation system. *Journal of Visual Languages and Computing* **7,** 109–127.
14. O. Losson & J. M. Vannobel (1998) Sign language formal description and synthesis. *International Journal of Virtual Reality* **3,** 27–34.
15. S. K. Liddell & R. E. Johnson (1989) American sign language: the phonological base. *Sign Language Studies* **64,** 195–277.
16. D. Tolani & N. I. Badler (1996) Real time inverse kinematics of the human arm. *Presence* **5,** 393–401.
17. R. Boulic, Z. Huang, N. Magnenat-Thalmann & D. Thalmann (1994) Goal-oriented design and correction of articulated figure motion with the TRACK system. *Computer & Graphics* **18,** 443–452.
18. N. I. Badler & D. M. Chi (1999) Virtual human animation based on movement observation and cognitive behavior models. *Proceedings of Computer Animation '99*, Geneva.
19. Witkin & Z. Popoci'c (1995) Motion warping. *Proceedings of SIGGRAPH '95*, pp. 105–107.
20. Bruderlin & L. Williams (1995) Motion signal-processing. *Proceedings of SIGGRAPH '95*, pp. 97–108.
21. C. Rose, B. Guenter, B. Bodenheimer & M. F. Cohen (1996) Efficient generation of motion transitions using spacetime constraints. *Proceedings of SIGGRAPH '96*, pp. 147–154.
22. J. Granieri, J. Crabtree & N. Badler (1995) Production and playback of human figure motion for 3D virtual environments. *Proceedings VRAIS '95 IEEE*, pp. 127–135.
23. K. Perlin (1995) Real time responsive animation with personality. *IEEE Visualization and Computer Graphics* **1,** 2–15.
24. D. Zeltzer & M. B. Johnson (1995) Motor planning: an architecture for specifying and controlling the behaviour of virtual actors. *The Journal of Visualization and Computer Animation.*
25. D. Zeltzer & S. Gaffron (1996) Task-level interaction with virtual environments and virtual actors. *International Journal of Human–Computer Interaction* **8,** 73–94.
26. T. Noma & N. I. Badler (1997) A virtual human presenter. *Animated Interface Agents Workshop*, IJCAI '97.
27. S. Kita, I. Van Gijn & H. van der Hulst (1997) Movement phases in signs and co-speech gestures, and their transcription by human coders. In: *Gesture and Sign Language in Human–Computer Interaction*, Springer-Verlag, Berlin, pp. 23–34.
28. C. Cuxac (1983) Autour de la Langue des Signes. *Journées d' études* **10**.
29. B. Moody (1983) La langue des signes. Dictionnaire bilingue élémentaire. *Ellipses* **1** & **2**.
30. Braffort (1996) *Reconnaissance et compréhension de gestes, application à la langue des signes*. Ph.D. thesis, University of Paris XI-Orsay, France.
31. D. Hernandez (1994) Qualitative representation of spatial knowledge. *Lectures notes in Artificial Intelligence*, Vol. 80, Springer-Verlag, Berlin.
32. S. Gibet & P. F. Marteau (1994) A self-organized model for the control, planning and learning of non-linear multi-dimensional system using a sensory feedback. *Journal of Applied Intelligence* **4,** 337–348.
33. S. Gibet & T. Lebourque (1994) Automatic motion control. *Proceedings of Eurographics '94*, Oslo.

34. T. Lebourque & S. Gibet (1999) High level specification and control of communication gestures: the GESSYCA System. *Proceedings of Computer Animation '99*, Geneva.
35. W. L. Zagler, B. Geoffrey & R. R. Wagner (1994) Notational representation of sign language: a structural description of hand configuration. In: Lecture Notes in Computer Science. Computer for Handicapped Persons. In: *Proceedings of the 4th International Conference ICCHP '94*, Springer-Verlag, Berlin, pp. 38–45.
36. P. M. Fitt's (1954) The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* **47,** 381–391.
37. P. Viviani & M. Cenzato (1985) Segmentation and coupling in complex movements. *Journal of Experimental Psychology: Human Perception and Performance* **11,** 828–845.