Gesture Modeling and Animation Based on a Probabilistic Recreation of Speaker Style

MICHAEL NEFF
University of California, Davis

MICHAEL KIPP
DFKI - German Research Center for Artificial Intelligence

IRENE ALBRECHT
MPI Informatik

HANS-PETER SEIDEL
MPI Informatik

Animated characters that move and gesticulate appropriately with spoken text are useful in a wide range of applications. Unfortunately, this class of movement is very difficult to generate, even more so when a unique, individual movement style is required. We present a system that, with a focus on arm gestures, is capable of producing full-body gesture animation for given input text in the style of a particular performer. Our process starts with video of a person whose gesturing style we wish to animate. A tool-assisted annotation process is performed on the video, from which a statistical model of the person's particular gesturing style is built. Using this model and input text tagged with theme, rheme and focus, our generation algorithm creates a gesture script. As opposed to isolated singleton gestures, our gesture script specifies a stream of continuous gestures coordinated with speech. This script is passed to an animation system, which enhances the gesture description with additional detail. It then generates either kinematic or physically simulated motion based on this description. The system is capable of generating gesture animations for novel text that are consistent with a given performer's style, as was successfully validated in an empirical user study.

Categories and Subject Descriptors: I.3.7 [**Computer Graphics**]: Animation

Additional Key Words and Phrases: human modeling, gesture, character animation

---

1. INTRODUCTION

People are engaged by characters with interesting personalities. However, creating quality animation for generic characters that correctly coordinates appropriate gestures with spoken text is already a challenging task. Generating movement that reflects a particular personality significantly increases the challenge of the gesture animation task, yet it is a goal towards which we must strive. This work describes one approach towards that goal. We present a system that allows the gesturing pattern of specific individuals to be modelled and then generates animation for new text from this model, complete with appropriate gestures and body movement that reflect the original subject. The movement

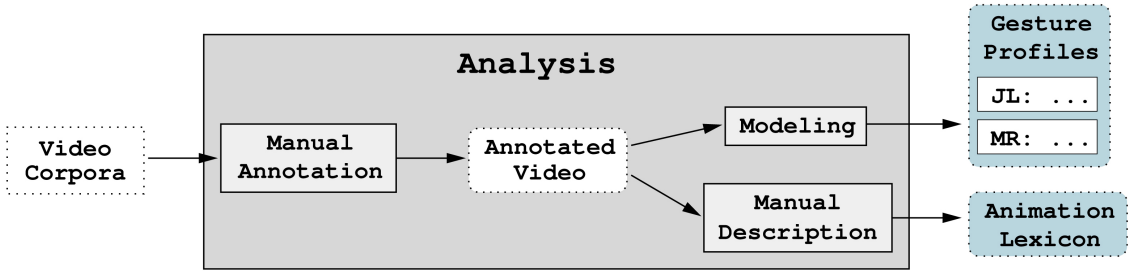

Fig. 1. Preprocessing phase of the system

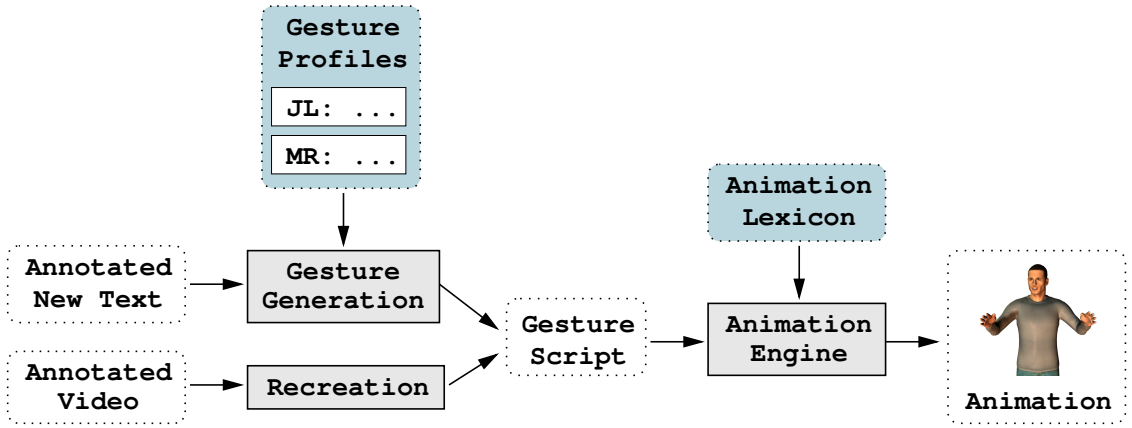

Fig. 2. Automatic generation process.

focus is on standing characters performing arm gestures, but the approach is also applicable to the entire body.

The system operates in two phases: a preprocessing phase and a fully automatic generation phase. The preprocessing stage is shown in Figure 1. Producing animation of a particular individual begins by collecting a video corpus for that person. In our case, we used two talk show hosts as subjects, employing about ten minutes of film of each. During an analysis step, the video is hand annotated in the tool ANVIL [Kipp 2001], and both gesture and more general animation data are extracted. A statistical model called a *Gesture Profile* is built based on the annotated data. In addition, an *Animation Lexicon* is constructed that contains data such as the normal hand orientation for each gesture that we model. These two components provide the input for the gesture generation and animation stage. This tool-supported analysis step allows us to generate a particularly wide range of gestures.

Once the initial analysis phase is complete, and any novel input text is tagged, the generation process is fully automatic, as summarized in Figure 2. There are two paths in this pipeline. The bottom path shows *re-creation*. Here, the annotated data from the video corpus is mapped directly to a *gesture script*, which is then animated. Re-creation can be used to produce animations of any annotated segment in the corpus. This is useful for validating the annotation and creating an animation of a specific performance. The second path in the pipeline generates animation for any *novel* tagged text[1], which need not be in the video corpus. The gesture script is generated from the statistical model for

[1]In addition to linguistic data, the text input must include timestamps for beginning and end of each word. This data is usually provided by the text-to-speech (TTS) software.

the individual specified by the user. This gesture script is passed to the *Animation Engine*, which further refines the description of the motion, using data from the animation lexicon and a set of rules described below. The animation engine produces final animated output either kinematically, or using dynamic simulation, at the user's option.

Traditionally, modeling gesture production and gesture animation are handled by well separated systems. For instance, the gesture specification system might only produce a gesture name, e.g. "beat", that is then rendered by the animation system by playing a pre-existing clip. In this work, we take a more tightly coupled approach that raises interesting issues of data representation and flow: What data gets generated where? Some of the detailed information needed to animate a gesture is best stored as part of the gesture model, i.e. on the generation side. This information can be related to the model of a particular performer, the correlations within a stream of generated gestures, synchronization with speech, or be definitional information for a particular gesture. Modeling this data as part of the generation process allows greater control for the gesture generation system and reduces the burden of how much the animation system needs to "know" and model. It frees the animation system from needing to access the gesture model. At the same time, representing data on the generation side can be costly in terms of modeling effort as it requires sufficient video footage, annotation work and the construction of statistical or rule based models. In keeping with principles of data encapsulation, we also wish to minimize what the gesture system needs to know about the animation system. To negotiate this trade off, our gesture generation system produces sparse data that captures the key definitional aspects of a gesture and provides good control. The details of the motion are filled out by the animation system, which also has additional information about the figure being modeled, gesture types and the controls available. Specifically, the animation system will complete timing information, deal with spatial conflicts and add in a more rich description of gesture form as it augments the sparse data it receives from the generation stage.

In comparing our system to a motion capture approach, as well as noting the greater control afforded the generation engine in our approach, it is worth examining the range of gestures our system can produce. We currently model 28 different gestures in our animation lexicon, each of which can be generated with either or both hands in any of hundreds of spatial locations and with an arbitrary number of after-strokes (Section 3). Thousands of different combinations are possible and the animation lexicon can be easily extended. Developing an appropriate motion capture database to cover this space would be a daunting, if not prohibitive, task. Another strength of our technique is that it can be used on any subjects for whom there is adequate film of them gesturing. This means that it can potentially be used on subjects that are no longer alive or who cannot otherwise be motion captured due to cost or availability.

Two talk shows hosts, JL (Jay Leno) and MR (Marcel Reich-Ranicki), are used as subjects in this paper. They have different gesturing styles (e.g. Figure 3) and also speak different languages. This illustrates an important point: our technique can be used to create gestural animation for text in languages different to that spoken by the subject. In our animation system, we employ a skeleton model containing 89 degrees of freedom, including six degrees for world space orientation and location and 21 degrees of freedom for each hand.

In our approach to gesture synthesis we bring together a number of features that have already been studied previously but not received this level of attention:

— Determination of gesture phase structure.

— Precise timing for multiple strokes.

— Inclusion of the gesture's spatial structure (location in space, curvature in space, hand and wrist shape, arm inclination).

— Developing a broad range of gestures.

— Controller-based physical simulation of motion for gesturing.

Truly novel contributions of this paper include:

— The selection and customization of gestures based on speaker-specific statistical models. The "recognizability" of the generated speaker-specific styles was confirmed with a user study.

— The development of data models at an appropriate level-of-detail to encapsulate sufficient data for speaker-specific re-creation and effectively divide the modeling task between gesture generation and animation components, while minimizing the annotation task.

— The production of contiguous gesture sequences, so-called *g-units*, instead of singleton gestures.

— Joint space approaches for controlling motion curvature and generating progressives.

— The synchronization of dynamic character simulation with speech production.

2. BACKGROUND

To generate and animate gestures from an input of tagged text is a fairly recent endeavor, pursued in an interdisciplinary arena requiring competencies from computer animation, artificial intelligence and psychology. Cassell *et al.* [1994] developed a rule-based system that generates audiovisual speech, intonation, facial expression, and gesture by working on the input text's information structure which is still common practice today. Another common practice is to synchronize the gesture stroke to the accented syllable of the coexpressive word, although, as we will show, it makes sense to sometimes synchronize the stroke with a different part of the sentence. Using the same agent as Cassell *et al.*, Noma *et al.* [2000] built the Virtual Presenter where gestures can be added to a text manually or with keyword-triggered rules. Animated gestures are synchronized with the following word. While the number of possible gestures is very small the focus was on how to implement meaningful rules from the literature on good public speaking. The system takes into account posture and eye contact with the audience. A more complex generation system is the Behavior Expression Animation Toolkit (BEAT) [Cassell et al. 2001]. It takes plain text as input and first runs a linguistic analysis on it before generating intonation, facial animation, and gestures. Gestures are generated using hand-made rules and are selected using priority values. While our system shares the overall goal of BEAT, to create accompanying gestures for a given text, there are a number of differences. In BEAT, a gesture is basically a "black box" that is triggered by a hand-made rule. In contrast, our system triggers gestures probabilistically and plans both the gestures' internal structure (phases, timing, shape) and macro-structure (by creating so-called *gesture units* [Kendon 2004]). Moreover, we produce a wide range of 28 different gestures[2] while BEAT seems to focus on very few samples. Most importantly, our approach produces not only natural-looking animations but a *character-specific* gesture style that intends to capture the individual differences of human beings.

Stone *et al.* [Stone et al. 2004] also use a data-driven approach to re-create a specific person's gesturing style. They re-arrange pre-recorded chunks of audio and motion captured pieces of full-body movement. Possible sentences are defined by a simple grammar. The corresponding utterance is assembled from those speech phrases and gestures that match the communicative function and minimize the required amount of time warping and blending. However, the range of both possible utterances and gestures is limited to what has been pre-recorded. In contrast, in our approach we extract abstract models of behavior that are then used to create and animate gestural behaviour on totally new input. Other relevant animation systems include EMOTE [Chi et al. 2000] which presents a kinematic system for expressive variation of arm and torso movements that is based on the analysis of Laban. Hartmann et al. [2002] present a kinematic animation system that realizes a gesture language that they have developed. Our gesture representation

[2]This does not even take into account the large variation that we achieve with different positions and varying phase structure, especially multiple strokes.

shares many features in common with theirs. We extend their representation in several ways: an additional spatial dimension is modeled (swivel angle), both world space and local hand orientation constraints are supported, additional movement features like posture, trajectory and tension changes are modeled, and a more complex representation for after-strokes is developed. Hartmann *et al.* [2006] extend their system to add expressivity to their gesture synthesis by varying activation, spatial and temporal extent, fluidity and repetition. [Noot and Ruttkay 2004] is also concerned with gesturing style. A style consists of a dictionary of meaning-to-gesture mappings, motion characteristics, and modality preferences. Combining style dictionaries yields mappings for new cultural groups or individuals. In contrast to our approach, their styles are hand crafted and model the behaviour of stereotypic groups instead of real individuals. In addition, the placement and frequency of gestures is fully determined by tags in the input text, and gestures are modeled at a comparatively coarse level since the paper's focus is a style description language, and it is not concerned with animation issues.

Kopp *et al.* [004a; 004b] present a gesture animation system that makes use of neurophysiological research and generates iconic gestures from object descriptions and site plans when talking about spatial domains, e.g. giving directions. Iconic gestures resemble some semantic feature of an object referred to in the co-occurring speech. Recent work on sign language generation by Henerfauth *et al.* [2007] has looked at similar issues of spatial mapping in order to relate ASL expressions to locations in space. In contrast, in our approach the domains are mostly non-spatial. Many iconics that occur in everyday conversation are either metaphoric and therefore standardized [McNeill 1992] or verge on the emblematic (e.g. gestures for actions like drinking or counting) and thus also standardized. Therefore, Kopp *et al.*'s approach and ours can be considered complementary. More in line with our approach is de Ruiter's [2000] Sketch Model where both gesture and speech originate in the same module called a *conceptualizer*. Gestures are processed in data structures with unbound variables, so-called *sketches*, that can be filled according to context and using a gestuary of concept-to-shape entries. A *gesture planner* fills the remaining parameters like body part (which hand(s)) and spatial locations and builds a final motor program for the articulators. The model is not implemented but can predict certain phenomena in gesture-speech synchronization. Our approach shares the processing of underspecified gesture structures which we call gesture frames.

Neff and Fiume [2005] present a system for modelling gesture-like movements using physical simulation, but do not model a complicated range of gestures or combine them with speech. Physical simulation has been used for many years to generate character motion with two main approaches emerging: optimization techniques that use physical laws as constraints [Witkin and Kass 1988; Popovic and Witkin 1999] and simulation techniques that forward simulate Newton's laws to generate motion [Hodgins et al. 1995]. We take a simulation approach, and in particular, follow on work in hand-tuned control [Hodgins et al. 1995; Faloutsos et al. 2001] where a proportional derivative (PD) controller is used at each character Degree of Freedom (DOF) to generate the required torques to make it move. As we have an underlying kinematic motion representation, our approach is also similar to the use of physical control to track motion capture data presented by [Zordan and Hodgins 2002], and our hand model is similar to [Pollard and Zordan 2005]. To our knowledge, this is the first use of forward simulation on a character of this complexity that must synchronize its movements with speech.

3. UNDERSTANDING GESTURE

A good way to approach a concept as diffuse and organic as gestures is to look at their temporal structure which can be nicely described in terms of phases, phrases and units [McNeill 1992; Kita et al. 1998; Kendon 2004; McNeill 2005]. A single *gesture* can be described as consisting of a number of consecutive movement phases. This can be expressed by the following rule[3]:

[3]Nonterminals are set in smallcaps, terminals in boldface, and optional elements are put in square brackets.

$$\text{GESTURE} \rightarrow [\ \textbf{preparation}\ ]\ [\ \textbf{hold}\ ]\ \text{STROKE}\ [\ \textbf{hold}\ ] \tag{1}$$

Only the stroke phase must occur in every gesture, all other phases are optional. The stroke is the "most energetic" and "meaning-carrying" phase of the gesture while in the preparation phase the hands are moved to the stroke's start position. The *hold* phases before and after the stroke are optional pauses, usually interpreted as a means to correctly synchronize the stroke with accompanying speech. The stroke can consist of multiple repeated movements which would make it a *multi-stroke*[4]. Since the first stroke in a multi-stroke is often the most pronounced and the following strokes have similar form but look weaker than the first, we call the first stroke the *main stroke* and all subsequent strokes *after-strokes*:

$$\text{STROKE} \rightarrow \textbf{main\_stroke}\ (\textbf{after\_stroke})^{*} \tag{2}$$

After-strokes are almost like small gestures themselves, each with their own preparation-stroke-hold structure. This becomes relevant in actual animation as elaborated in Section 6.5.

The complete GESTURE is also called a *gesture phrase* (g-phrase) in the literature. Opposing McNeill's claim that every gesture has a stroke, Kita *et al.* [1998] found that some gestures have a single meaningful still phase instead, called an independent hold[5]. Imagine the prototypical "raised index finger" where the hand is raised, held still and retracted: instead of an energetic stroke there is only a single hold phase, therefore called an independent hold. We distinguish two principal gesture types, stroke gestures (S-GESTURE) and hold gestures (H-GESTURE), and expand rule (1) to the following three rules:

$$\text{GESTURE} \rightarrow \{\ \text{S-GESTURE} \mid \text{H-GESTURE}\ \} \tag{3}$$

$$\text{S-GESTURE} \rightarrow [\ \textbf{preparation}\ ]\ [\ \textbf{hold}\ ]\ \text{STROKE}\ [\ \textbf{hold}\ ] \tag{4}$$

$$\text{H-GESTURE} \rightarrow [\ \textbf{preparation}\ ]\ \textbf{hold} \tag{5}$$

We call a *rest position* a pose where the hands either hang down at the side or are supported in some way: e.g., arms lie on an arm rest, arms are folded, hands are in pockets or are locked behind one's back. A gestural excursion always starts from a rest position, can encompass one or more gestures and finally returns to a rest position. Such an excursion is called a *gesture unit* (g-unit). For gesture generation, the g-unit is an important organizational entity as it groups together multiple gestures in one continuous flow of movement. A unit always ends with a *retraction* movement to a rest position.

$$\text{UNIT} \rightarrow (\text{GESTURE})^{+}\ \textbf{retraction} \tag{6}$$

A unit can consist of a single gesture. McNeill [1992] actually found that his subjects frequently perform only one gesture per unit (only 44% of the time would his subjects perform more than one gesture per unit). However, his subjects consisted of people who were neither trained nor experienced in speaking in public or on TV. In contrast, our data of professional TV performers shows a completely different picture. Table I shows how often the different g-unit

[4]Kita *et al.* calls them *multiple strokes*, Hartmann *et al.* [2005] calls them *repeats*.
[5]In his latest book, McNeill [2005] acknowledges the existence of independent hold but calls them *stroke holds*.

sizes occured. Our speakers frequently combine multiple gestures to units: MR uses units with more than one gesture 66.7% of the time, JL 64.3% of the time. We believe that this is one reason why JL's and MR's gestures are enjoyable to watch: the speakers produce a fluent stream of continuous gestures instead of isolated singleton gestures. One aim of our project was to transfer this quality to synthetic agents. In follow-up work [Kipp et al. 2007], we have validated our hypothesis, demonstrating that the the use of gesture units rather than singleton gestures is perceived to be more natural, more friendly and more trustworthy. Consistent with this, the use of singleton gestures was perceived to show greater nervousness.

	1	2	3	4	5	6	>6	total number of units
JL	35.7 %	15.7 %	17.1 %	5.7 %	11.4 %	5.7 %	8.6 %	70
MR	33.3 %	16.7 %	11.1 %	14.8 %	9.3 %	3.7 %	11.1 %	54
McNeill	56 %	14 %	8 %	8 %	4 %	2 %	8 %	254

Table I. Number of gestures per unit for our speakers JL and MR in comparison with McNeill's subjects. The table shows that JL and MR use units with more than one gesture much more often than McNeill's subjects.

3.1 Gesture Lexicon and Lexemes

It follows from the encountered usage patterns that since we want to produce gesture behaviour that looks characteristic for a certain person, we need to produce a *broad* spectrum of gestures. Previous work focussed on a limited range of specific gestures in order to work out details, e.g., about the semantics-form relationship between speech and iconic gesture [Kopp et al. 004b]. However, in our target domain, iconic gestures that need a deep understanding of semantics and form rarely occur.

In everyday conversations, but also in talk shows and formal presentations, human speakers use mostly gestures where no strong semantic function is visible. McNeill calls these gestures *metaphorics* since the relationship between the gesture and what is said is only etablished through an abstract metaphor. For instance, in a *progressive* gesture the speaker's hands revolve around each other in circles. McNeill argues that the gesture refers to the abstract notion of a forward rotating wheel which in turn refers to a co-occuring word in speech like, e.g., "going", "developing" or even "future". These are the gestures we focus on. However, do these gestures share a common form or is their shape totally arbitrary and invented on the fly?

While it is common knowledge that emblematic gestures (e.g. the victory sign or the thumbs-up gesture) are drawn from a shared, though culture-specific, lexicon, it became clear only recently that this is also true for more abstract gestures. Webb [1997] showed for a number of speakers that they use metaphoric gestures from a shared lexicon of forms (see also [Kipp 2004]). Although each speaker applies slight variations and only uses a subset of these gestures, there is basically one large reservoire of gestures that all speakers draw from[6]. In our approach to gesture generation we exploit this insight to represent gestures as lexicon entries, so-called *lexemes*, which can be considered equivalence classes with respect to form and function.

Kipp [2004] collected a lexicon of gestures for two German TV show hosts. To this work, we added a new speaker with

[6]Such common reservoires of gestures may be culture-specific. While intercultural differences are well explored for emblematic gestures (cf. [Axtell 1998], [Saitz and Cervenka 1972]), there is only sparse literature about this aspect for more abstract gestures like metaphorics (cf. [McNeill 2005], [Calbris 1990]).

a different language: the American talk show host JL. We assembled a gesture lexicon of 39 lexemes[7] and annotated a video corpus (Section 4). Of this gesture lexicon, MR uses a subset of 31 lexemes and JL uses 35. The large overlap of 27 lexemes that both MR and JL use supports the hypothesis of a shared lexicon of gestures that all people use. Figure 3 shows sample frames for the most frequently occurring gesture lexemes. Some gestures are depicted with two frames, taken at the beginning and end of the stroke, others with a single frame of JL and MR each performing the same lexeme.

JL		MR	
lexeme	*%*	*lexeme*	*%*
Cup	24.4	Cup	6.9
PointingAbstract	8.9	RaisedIndexfinger	6.9
PointingPerson	6.7	FlingDown	5.8
HandClap	6.7	Wipe	5.8
Shrug	6.2	Beat	5.3
Progressive	4.4	Calm	5.3

Table II. The table shows the six most frequently used lexemes for each speaker and how often the particular lexemes are used (in %). Figure 3 illustrates all of these particular lexemes, except for PointingPerson, HandClap and Shrug which have self-explaining labels.

While the gesture lexicon represents what is *shared* between speakers, the specific subset that each speaker uses and the frequency of each lexeme are significant aspects for modeling *interpersonal differences*. As Table II shows, the speakers differ significantly in what lexemes they use and how often they perform a particular lexeme. As we will show in the following section, we further model the variations of gesture form between each speaker and generate particular lexemes in correlation with a speaker's tendency to use those lexemes for a given speech segment.

4. ANALYSIS

To automatically generate and animate gestures in a speaker-specific style, a human speaker has first to be studied and analyzed. Using a combination of manual labour and automatic data extraction, the key factors of speaker gesture behaviour are then stored in machine-readable form for automatic gesture generation and animation.

Figure 4 gives a schematic overview of the analysis process. First, a video corpus for each speaker is annotated by hand. Both speech[8] and gestures are transcribed by human coders. This annotated corpus is used for three purposes. First, all annotated gestures are stored in the GestureDB database, as templates for automatic generation. Second, key properties for each gesture lexeme, especially the relationship between gesture and speech, are modeled with statistical tables. Both the GestureDB and the statistical model are stored in speaker-specific *gesture profiles*. Third, for animation, speaker-specific lexeme properties are modeled in the *animation lexicon* (Section 4.1.3).

4.1 Subject Annotation

The video annotation serves to translate essential concepts in the source video material into a machine-readable form. A human coder annotates linguistic entities (e.g. words) and gestural entities, including temporal boundaries, on tracks on an annotation board (Figure 5). In order to check how well the annotation captures what happened in the original

[7]Note that the animation engine currently only models a subset of 28 lexemes. This is partly due to rare lexemes that occur in the video corpus but were never generated in our examples because of their low probability and partly due to gestures whose shape must be determined by semantic knowledge not modeled by our system.

[8]Word and phoneme boundaries and timings are marked. At the minimum, phoneme information could be extracted automatically using current tools, e.g. CMU Sphinx.

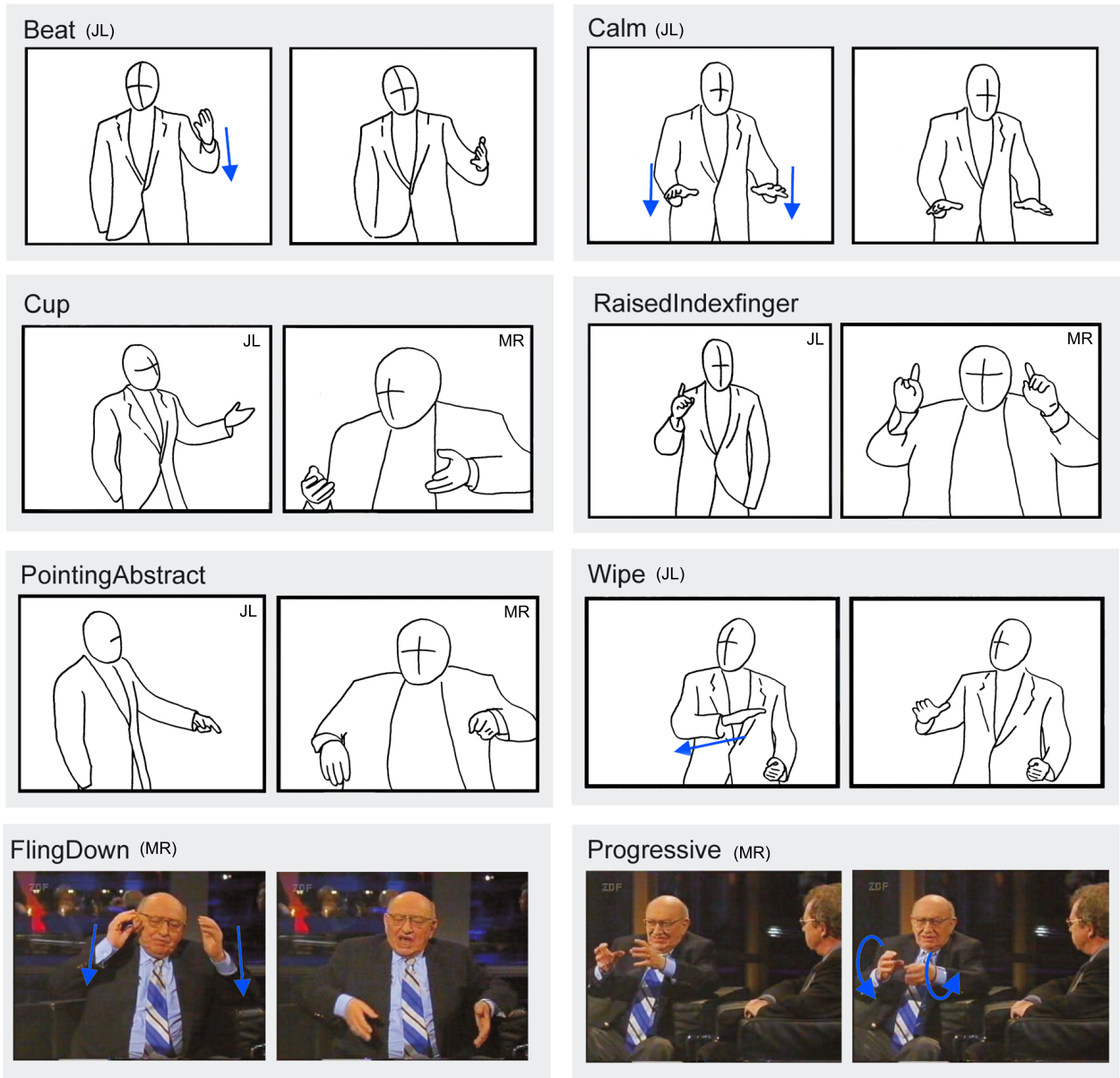

Fig. 3. Examples of eight frequently occurring gesture lexemes (cf. Table II). They are usually depicted with two frames from the same speaker, showing the beginning and end of the stroke (Beat, Calm, FlingDown, Progressive, Wipe). Cup, PointingAbstract and RaisedIndexfinger are illustrated with a single frame of JL and MR each, performing the same lexeme.

video we can feed it directly to the animation system, doing what we call a re-creation of the original behaviour (Figure 2). This is similar to what Frey [1999] called re-animation, and Martin et al. [2006] call copy-synthesis. The manual annotation is a work-intensive process: 1 minute of video takes about 90 minutes of coding by a human coder. However, coding can be done by anyone after a brief period of training; no special knowledge of animation or

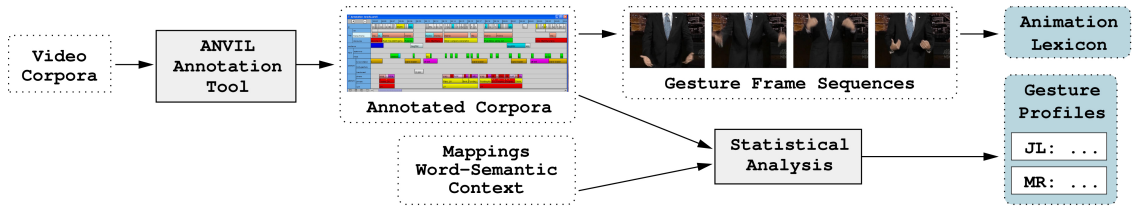

Fig. 4. Analysis pipeline.

linguistics is required. To support manual annotation, we use the video annotation tool ANVIL [Kipp 2001] and the phonetic analysis tool PRAAT [Boersma and Weenink 2005].

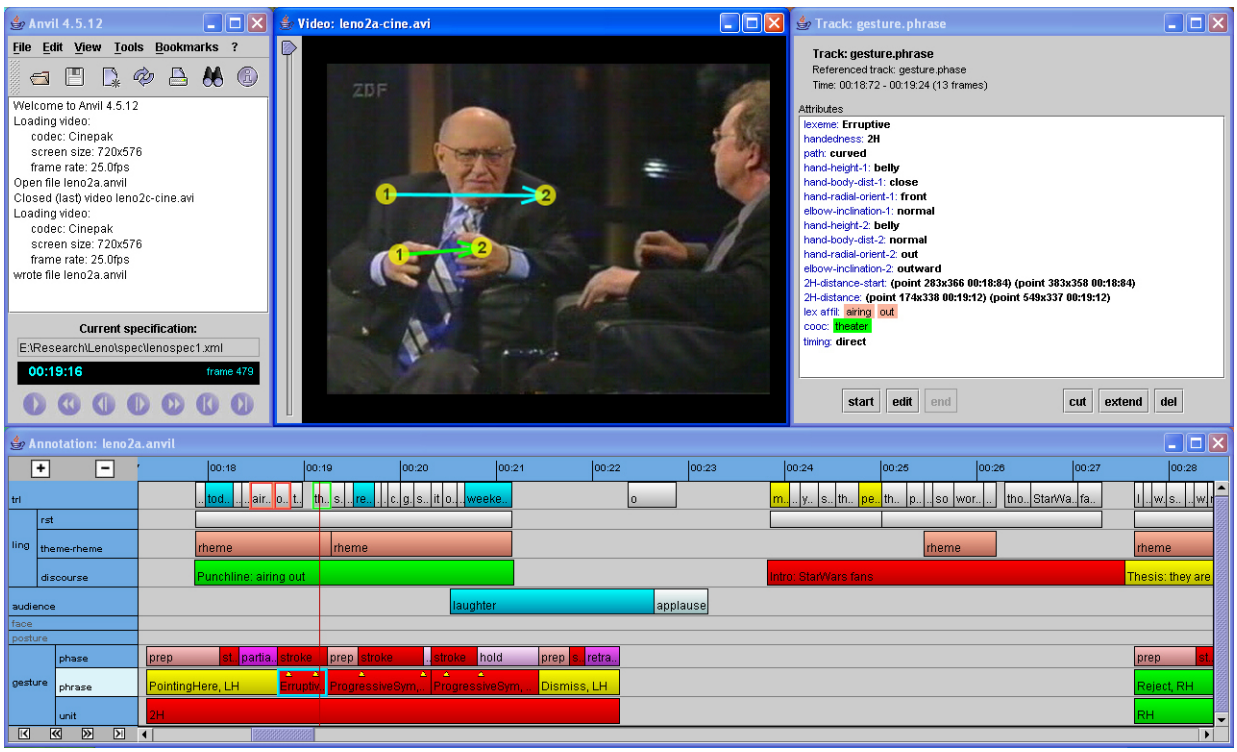

Fig. 5. The ANVIL video annotation tool allows human coders to efficiently encode time-aligned information for digital video. During analysis, a video corpus for each speaker is transcribed for statistical modeling of gesture behaviour. The bottom window contains the multi-track annotation board where coding takes place.

4.1.1 *Speech Annotation.* The linguistic part of the annotation consists of coding words, discourse segments and information structure. We use the PRAAT tool to perform a word-by-word orthographic transcription of the utterance, including the words' boundaries, which is imported to ANVIL. Words must be grouped into sentence-like units. We use clauses as defined by Rhetorical Structure Theory (RST) [Mann and Thompson 1988]. However, any kind of discourse segmentation works with our approach. Finally, information structure is annotated using the concepts of

theme, rheme and focus [Steedman 2000]. The theme is the part of the utterance that links the utterance to the previous discourse and specifies what the utterance is about, whereas the rheme relates to the theme and specifies something novel or interesting about it. Following Steedman [2000] we also annotate the focus, which is the part of the rheme or theme that distinguishes the rheme/theme from other alternatives the context makes available. We make the simplifying assumption that the emphasized word or phrase is the rheme's focus. For example:

During the battle [rebel spies managed to steal secret plans to the Empire's ultimate weapon the Death Star]$_{rheme}$

In the example the first three words refer to a battle that is introduced in the preceding sentence which makes it the *theme* of the utterance. The bracketed part, the *rheme*, introduces the new information. And since "the Death Star" is emphasized it is the *focus* of the rheme.

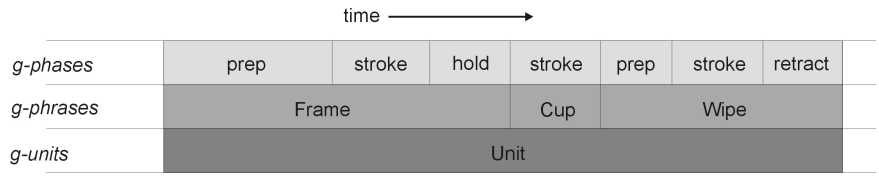

Fig. 6. Gesture annotation entities on three tracks.

4.1.2 *Gesture Annotation.* The gestural part of the annotation follows the hierarchical organization of gestures in phases, phrases and units, as described by rules (3)-(6) in Section 3. The human coder transcribes gestures in the video by adding *annotation elements* to three gesture annotation tracks in ANVIL. In the screenshot (Figure 5) these tracks (phase, phrase and unit) can be found at the bottom of the lower window; a schematic view is given in Figure 6. For each annotation element, the coder specifies begin and end times and then fills a number of attributes (attributes are displayed in the top right window in Figure 5). In the top track, called *phase*, gesture phases are transcribed following instructions by Kita *et al.* [1998]. The annotation elements contain one attribute for the phase type: preparation, stroke, hold, etc. The coder has a second attribute to specify the number of strokes if the phase is a multi-stroke.

On the second track, called *phrase*, several consecutive phases are combined into a gesture (e.g. Frame, Cup and Wipe in Figure 6). In Figure 5, the currently selected gesture, "Erruptive", is highlighted by a blue frame, and all its attributes are displayed in the top right window. Following instructions in Kipp [2004], we annotate the following attributes for each gesture: lexeme, handedness, lexical affiliate and co-occurrence (Table IV). The lexeme denotes the lexicon entry that the gesture corresponds to (e.g., Frame, Cup, Wipe). Handedness denotes the executing hand(s). The lexical affiliate is the word or phrase that corresponds to the meaning or function of the gesture [Schegloff 1984]: for instance, "he" or "this" for a pointing gesture or "driving" for a metaphoric progressive gesture. Since the lexical affiliate and the gesture do not always co-occur [McNeill 2005], the coder also specifies the word that the gesture co-occurs with. In Figure 5 the lexical affiliate to the current gesture is "airing out", highlighted with a red frame in the top track, while the co-occuring word "theater" is highlighted with a green frame in the top track.

We extended this scheme by adding information about the shape of the gesture (cf. [Kipp et al. 2006]). For each gesture, the coder specifies the trajectory (curved or straight) and hand/arm positions at the beginning and end of the stroke (s-gesture) or at the beginning of the independent hold (h-gesture).

Each hand/arm position is specified by a 4-vector $p = (h,d,r,s)$ for height, hand-body distance, radial zone and arm swivel angle. Each dimension of p has 5-7 discrete values (Table III and IV). For bihanded gestures, we additionally specify the hand separation (see Figure 5). Although hand separation could be computed from the position data (radial zone), the resulting range of values would be very small and it would be difficult to decide whether the hands touch.

Height	*Distance*	*Radial Orientation*	*Elbow Inclination*
above head	far	far out	orthogonal
head	normal	out	out
shoulder	close	side	normal
chest	touch	front	touch
abdomen		inward	
belt			
below belt			

Table III. Our three dimensions for hand position and one dimension for elbow inclination are divided into discrete intervals for annotation.

For hand separation annotation, we extended ANVIL: coders can edit 2D points on the video screen and store these points in an annotation element. Hand separation is annotated with two points, located at the middle of each palm, in the gesture phrase annotation element (middle layer in Figure 6). The shoulder width is also encoded and used to normalize hand separation.

Since this data is expensive to annotate we devised a minimal coding scheme that is sufficient to re-create the original gesture to a reasonable degree. More precise approaches to transcribing positional features (e.g. [Frey 1999] or [Martell 2004]) would increase the annotation effort by a factor of 2-10.

Note that we do *not* encode handshape and palm orientation in the manual annotation process. Palm orientation is assumed to be equal for each lexeme and is thus encoded in the Animation Lexicon (Section 4.1.3). Handshape is heuristically generated at runtime (Section 5.2.2) by selecting from a range of legal hand shapes which is pre-defined for each lexeme.

Encoded property	*Encoding*	*Description*
lexeme	{Cup, RaisedIndexfinger, Wipe, Progressive, ...}	Name of the gesture in the shared lexicon of conversational gestures.
handedness	{LH, RH, 2H}	Hand which performed the gesture (LH/RH), or both (2H).
lexical affiliate	*link to word(s) in the speech track*	The word or phrase in speech that semantically corresponds to the gesture, e.g. "you" for a deictic gesture to the addressee, or "going" for the gesture "Progressive".
co-occurrence	*link to word(s) in the speech track*	The words that temporally co-occur with the stroke of the gesture.
trajectory	{straight, curved}	Whether the gesture trajectory is straight or curved.
location 1	4-vector (height, hand-body distance, radial zone, arm swivel)	Location of the hand(s) at the beginning of the stroke or independent hold.
location 2	4-vector (height, hand-body distance, radial zone, arm swivel)	Location of the hand(s) at the end of the stroke. *Only for s-gestures.*
shoulder width	screen distance	Width of the shoulders in the current frame (used for normalizing hand separation). *Only for 2H gestures.*
hand separation 1	screen distance	Hand separation at the beginning of the stroke or independent hold. *Only for 2H gestures. Empty if view angle unsuitable.*
hand separation 2	screen distance	Hand separation at the end of the stroke. *Only for 2H s-gestures. Empty if h-gesture or view angle unsuitable.*

Table IV. The human coder specifies a number of properties for every gesture in the video. The g-phrase annotation element captures the gesture's semantic and temporal relation to speech and its form and development in space.

On the third annotation track, the coder groups together contiguous gestures, i.e. they are not interrupted by a full retraction, to a single unit. Every unit thus ends with a full retraction unless the video ends in mid-gesture. The unit element also stores the retraction position of the unit's last gesture (e.g., hands at side or hands clasped).

speaker	*duration*	*#phases*	*#gestures*	*#units*
JL	9:04	574	229	70
MR	8:31	496	192	54

Table V. The size of the annotated corpus for speakers JL and MR.

Data	*Description*	*Frequency of Use*
Hand Orientation	Constraints on wrist angles and/or forearm rotation expressed in either the local or chest frame. Can be specified for the start and end of a pose.	100%
Posture Parameters	Curvature to the spine or offset to the collarbones based on the parameterization presented in [Neff and Fiume 2006]. Balance and pelvic tilt adjustments may also be included.	17% collar, 20% spine
Progressive	Controls the scale of forearm and wrist rotation during a progressive.	Only used with progressives and regressives.
Tension Changes	Change tension to high, medium or low for a given joint either at the start or during a stroke.	not currently used
Transition warps	Warps the timing of the transition from the start pose to end pose of a stroke.	10%
Multi-stroke form	Can be specified for a stroke or prep and stroke phase. Data includes a vertical and horizontal offset to hand position, change to hand and forearm rotation and an offset in elbow bend.	43%
Multi-stroke timing	Percentage of time that should be used for a stroke and for a hold in a prep-stroke-hold afterstroke.	13%

Table VI. Parameters that can be defined as part of the animation lexicon. The frequency field indicates what portion of lexicon entries include each type of data for the corpus used in this paper.

In Table V we show the size and contents of the annotated corpus for the two speakers JL and MR. Both corpora are of similar size. It is also interesting that both speakers seem to have a similar gesture frequency, since speakers can differ noticeably in that respect [Kipp 2004].

4.1.3 *Producing an Animation Lexicon.* The animation lexicon is created as part of the annotation process and records additional information for each gesture lexeme. Whereas the previous annotations recorded data for each gesture sample in the corpus, only one entry is made in the animation lexicon for each gesture lexeme, irregardless of how many times a lexeme occurs in the corpus. This reduces the total annotation effort required. The data that is included in the lexicon is summarized in Table VI and will be described in more detail below. As can be noted from the table, data items are only added when needed and for most lexemes only a partial set of data is required.

The animation lexicon contains three main types of data: hand orientation, torso posture and data for after-strokes. Hand orientation is specified by rotation around the forearm and two rotational degrees of freedom in the wrist. These values can be specified either as joint angles or as constraints to be met in either world space or the character's chest space (the latter moves with the character and often proves more natural than a world space constraint). This information is definitional for almost all gestures and was recorded for every lexeme. As an example, a "cup" or a "shrug" will always have the palm facing up whereas a "dismiss" will have the palm facing down and end with a bent wrist.

Posture data includes spine and collar bone movements that are either definitional for the gesture or characteristic of the particular character. For instance, the chest will normally be opened (backward movement of the collar bones) during a "wipe". JL breaks with normal convention and in our data does not raise his shoulders during a "shrug". We use a reduced DOF posture parameterization based on [Neff and Fiume 2006] to represent this data. Specifically, the shape and intensity of spinal curvature in the coronal and sagittal planes can be specified, along with the amount of axial

rotation; spinal rotation can be specified; collar bones can be moved up or down and forward or back, either in parallel or opposition; weight shifts, moving the arm out from the character's side and pelvic twists are also possible. Thirty percent of our lexemes included some posture data and in cases such as a "wipe", it can be very important. Multiple possible posture changes can be specified for a given lexeme. At runtime, one of these will be chosen randomly without reference to other data defining the specific lexeme. Modeling correlations between posture changes and particular instances of a lexeme would likely be a beneficial addition in some cases, but would also increase the annotation overhead.

Recall that after-strokes are the small stroke repetitions following the main one in a multi-stroke. They carry similar meaning, but may differ in form and extent from the main stroke. They are generally smaller in amplitude and confined more to wrist and forearm movement. The prep and stroke data for these movements consists of forearm rotation, hand rotation, vertical or horizontal positional offsets, and elbow bend offsets. One prep and stroke are optionally specified to define each after-stroke and they are then repeated for each repetition.

The lexicon also includes additional data that can be definitional for certain gestures, such as warps to transition curves to change the timing profile, and amplitude values for progressives. As an example of such a change, a wipe gesture in which the hands are moved from the centre out to the side will generally feature an acceleration throughout the movement and will not look correct with an ease-in ease-out transition. Such changes to the transition can be achieved by specifying in the animation lexicon either a warping of the interpolation curve or a change in joint tension.

Authoring of the animation lexicon begins with images of each lexeme that serve as reference material. These images are automatically generated by the ANVIL annotation tool as shown in Figure 4. For gestures with straight trajectories, these images show the start and end pose of the stroke. For curved trajectory gestures, two internal frames are also generated. The annotation process is straighforward: the annotator simply examines the images for features that should be recorded in the animation lexicon (palm orientation, posture changes etc.) and then adds the corresponding data. If there are significant differences in multiple lexeme examples, it is possible to specify multiple versions of a lexeme with weights and one will be chosen randomly at runtime. In practice, this is only done when some example lexemes have a strong posture change that would become noticeable if it was repeated identically each time the lexeme was triggered. No particular training is required to perform the annotation, but a good ability to observe movement variation is an asset. In our experience it takes about a minute to a few minutes to annotate one gesture.

Animation lexicons are character specific, but we found for our two characters that most of the data from one lexicon can be used directly in the second. Posture variations appear to be the data most related to one of our subjects, and hence are more likely to be customized. The difference may be influenced by the fact that one subject is seated, but this alone does not seem sufficient to account for the variations observed.

4.2 Building a Gesture Profile

The annotated corpus is used to build a profile for the speaker's gesture behaviour. The profile consists of the sample database, GestureDB, a statistical model and average values. For the GestureDB, the annotated information for each gesture in the corpus is stored as a reproducible "gesture sample" of the specific speaker. These samples can be seen as high-level movement patterns that can be easily modified in a meaningful way.

The statistical model is automatically computed from the annotations. It models estimated probabilities and is used in generation to trigger gestures, to predict where they are placed relative to speech, and to determine parameters like handedness and frequency. To build the model, the speech transcription needs a two-step preprocessing. In a first step, morphological analysis maps words to their lemma (e.g. striking→strike, won→win).

In a second step, phrases, consisting of lemmas and/or words, are mapped to *semantic tags*. These tags abstract away

from the speech surface structure and partially capture aspects of semantics and communicative function. In generation they are responsible for triggering gesture candidates, based on the assumption that similar gestural forms can express the meaning of the subsumed words. We use a total set of 87 semantic tags. A subset of 28 frequent semantic tags together with samples from the corpus is shown in Table VII. In our approach we employ look-up tables both for morphological analysis and semantic tagging. However, both tasks could be automated using off-the-shelf software[9] or semi-automatic approaches[10].

For gesture generation we want to know how frequently the modeled speaker uses a particular gesture lexeme in conjunction with a particular semantic tag. We take the lexical affiliate annotations to estimate the conditional probability of gesture lexeme l occurring with semantic tag s over our corpus C=(G, S) consisting of all occurring gestures G and semantic tags S by

$$\hat{P}(l|s) = \frac{\#\{g \in G : \text{lexeme}(g) = l \wedge \text{lexaffil}(g) = s\}}{\#S}$$

These values define a probabilistic mapping from semantic tags to lexemes: For each semantic tag s we obtain many rules $s \rightarrow l_i$ with a "confidence value" $\hat{P}(l_i|s) \in [0,1]$. Since the semantic tags are language-independent the resulting gesture profiles can be used for any target language; we use them for German and English. We also store bigram models $P(l_i|l_{i-1})$ for lexeme sequences, i.e. the probability that lexeme l_i follows lexeme l_{i-1}. To model a speaker's preferred handedness and handedness shift patterns we utilize the unigram probability estimation $\hat{P}(h)$, the bigram estimation $\hat{P}(h_i|h_{i-1})$ and lexeme-relative handedness $\hat{P}(h|l)$, where $h \in \{LH, RH, 2H\}$. Observation showed that the way multiple strokes (i.e. repeated strokes) are used or not used can be very characteristic for a speaker. We therefore store the average number of strokes $\mu_{strokes}$ per lexeme. To model the timing offset between gesture and speech we also record the average time difference ΔT_{end} between end of word and end of stroke (for hold gestures we record the start time difference ΔT_{start}). Finally, on a higher level we record gesture rate which is the number of gestures per minute because the amount of gesture activity also seems to be quite characteristic for a given speaker.

5. GESTURE GENERATION

Once a speaker profile was created, our system can process any text input and produce an animation with accompanying gestures. This "runtime" system consists of two components: the NOVA[11] gesture generator, described in this section, and the animation engine, described in Section 6. NOVA processes the input text and produces a gesture script for the animation engine (Figure 7).

The input text must contain some additional information. For each word, the beginning and end time must be specified which is usually delivered by the text-to-speech (TTS) software. The text must also be segmented into utterances and contain information about the theme, rheme and focus of each utterance (Section 4.1.1). For gesture generation, the input is transformed to a graph structure which is then processed in four stages:

(1) Gesture creation

(2) Gesture selection

(3) Unit creation

(4) Unit planning

[9]For instance, MORPHIX for lemmatization [Finkler and Neumann 1988].
[10]For instance, WordNet [Miller et al. 1990] for semantic modeling as used in BEAT [Cassell et al. 2001].
[11]**NO**n**V**erbal **A**ction Generator

Semantic tag	*Words from the corpus*	*Semantic tag*	*Words from the corpus*
ADDRESSEE	'you'	NUMBER	'six' 'first' 'second' 'third'
AFTER	'after'	PERS_NAME	'President Bush' 'Zorro' 'Michael Jackson' 'Princess Leia'
AGGRESSION	'attack' 'crack down' 'civil war' 'strike' 'battle' 'weapon'	PERS_PRONOUN_OTHER	'they' 'he' 'she'
AGREEMENT	'yes'	PERS_PRONOUN_SELF	'I'
BEFORE	'last' 'ago'	POSSESSIVE_OTHER	'their' 'her'
CONJ_SEQ	'and'	POS_AFFECT	'encourage' 'win'
DEIC_HERE	'here' 'today' 'now'	PROCESS	'open' 'air out' 'send' 'practise' 'drink' 'create' 'manage' 'pursue' 'restore'
DEIC_THERE	'there'	QUEST_PART	'why'
DEMONSTRATIVE	'those' 'that' 'this'	QUEST_PART_PERS	'who'
DESTRUCT	'cancel' 'destroy'	REL_PRONOUN	'what'
DISTANCE	'long' 'far'	THERE_IS	'there is'
LOCOMOTION	'come' 'get in' 'go' 'go up' 'race'	TIME_POINT	'Tuesday' 'Monday' 'Wednesday'
NEGATION	'dont' 'not' 'no'	TITLE	'Star Wars' 'Revenge of the Sith' 'American Idol' 'galactic empire' 'Death Star'
NEG_EVAL	'wrong' 'horrible' 'sinister'	TOTALITY	'all' 'all you people' 'whole' 'entire'

Table VII. This table shows a subset of the semantic tags used and the corresponding words from the corpus (only the English ones). Lemmatized words are mapped onto these language-independent tags to achieve a semantic abstraction from the surface words.

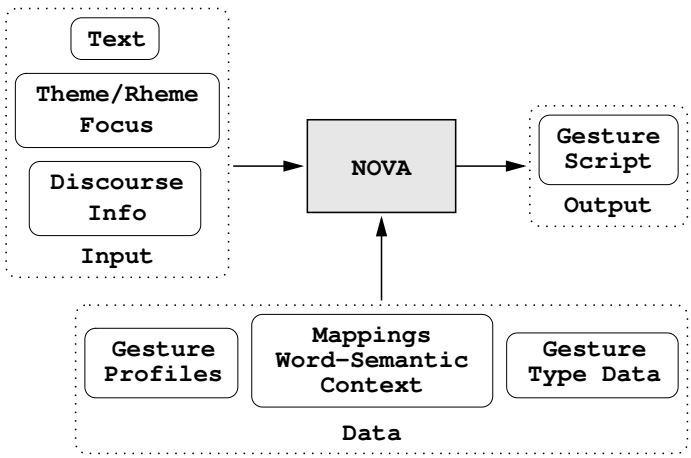

Fig. 7. The NOVA gesture script generator.

The output is written to a gesture script which contains character-specific gestures, organized in units, with locational and timing parameters for animation.

5.1 Gesture Frames and Generation Graph

To generate, select and plan the gestures we use a graph structure to represent both speech and gestures (Figure 8). Generated gestures are inserted as arcs in the graph and represented as feature structures, so-called *gesture frames*.

The underspecified frames become gradually enriched during generation using the speaker's profile and contextual constraints.

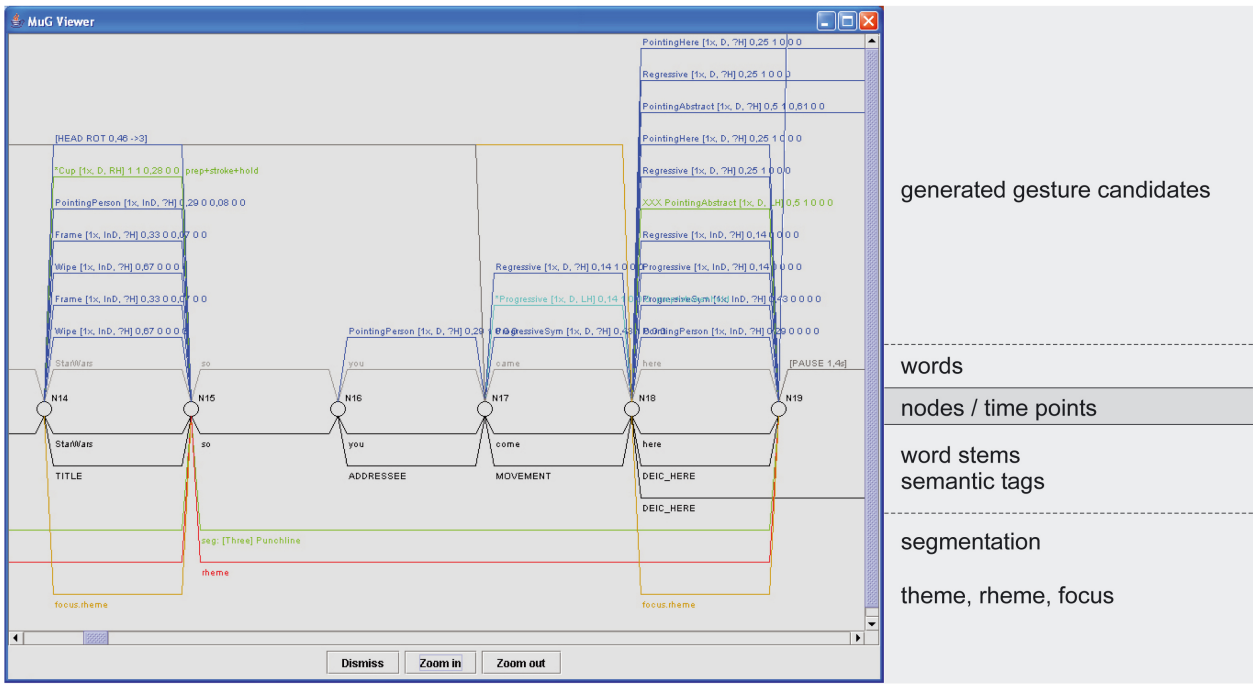

Fig. 8. Multimodal generation graph containing input text with theme/rheme/focus tags (lower arcs). Upper arcs represent generated gesture frames.

The input text is used to construct the initial graph. For each word, a node representing the begin time is created. While nodes represent time points, arcs represent concepts with a temporal duration: words, utterances, theme/rheme/focus. Words are mapped to lemmas that are added as arcs. Likewise, lemmas are mapped to semantic tags and added to the graph. Not every word has a semantic tag. With our current mapping we find semantic tags for 39% of the words.

5.2 Gesture Generation Algorithm

5.2.1 *Gesture Creation.* In the first step we produce many candidate gestures for the given text by adding gesture arcs to the graph. For this, we use the concept-to-gesture mapping from the speaker's gesture profile (Section 4.2). For each rheme ρ, for each semantic tag s in ρ, we produce an underspecified gesture frame of lexeme l iff $\hat{P}(l|s) > 0.1$. Additionally, we place a copy of this frame on the nearest *focus* within ρ. This simulates the phenomenon that gestures sometimes do not synchronize with their lexical affiliate like in "destroy an [entire planet]" where a Wipe gesture is performed on the bracketed part, although "destroy" is the lexical affiliate. The added gesture frame is represented by an arc that stretches across s, indicating the gesture's temporal position.

5.2.2 *Gesture Selection.* In the next step, we select candidates and specify handedness and handshape. A path of non-overlapping gestures is selected using the gesture rate to determine the desired number of gestures N. We then select the most likely sequence of gestures $(g_0, \ldots, g_{N-1})$ [Jurafsky and Martin 2003], i.e. the sequence that maximizes

$$\prod_{i=0}^{N-1} \left(0.6\hat{P}(g_i|s) + 0.4\hat{P}(g_i|g_{i-1})\right)$$

We model the sequence of gestures because of our observation that some speakers use idiosyncratic *combinations* of gestures. Reproducing these combinations is desirable. However, since our training data is sparse we reduce the bigram's weight. This might change with larger corpora. The next step determines the handedness of the gestures using a linear combination of estimated probabilities

$$\overline{P}(h_i|h_{i-1}) = 0.5\hat{P}(h_i|g_i) + 0.2\hat{P}(h_i) + 0.3\hat{P}(h_i|h_{i-1})$$

where the weights were empirically determined and the handedness is found by maximizing the probability of the handedness sequence $(h_0, \ldots, h_{N-1})$. This model captures the observation that speakers differ with respect to which hand(s) they prefer (LH, RH or 2H) and how often they change their "handedness mode". Figure 9 illustrates the handedness model. The area of the circles indicates the absolute (unigram) probability of a gesture being performed in this mode, while the arrows and numbers indicate the probability (in %) of switching from one mode into the other when going from one gesture to the next.

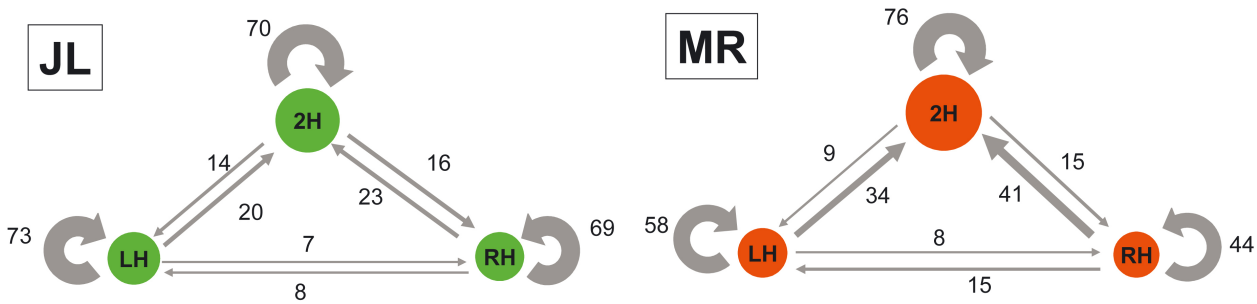

Fig. 9. Handedness transition diagrams for JL and MR show preferences which hand(s) are used for gesturing and how often this mode is switched. Circle area indicates unigram probability, size of the arrows and number indicate transition probability between gestures. The diagrams show that MR uses 2H more often than JL. Moreover, JL stays in one mode more often than MR, as the high probabilities on the 2H→2H, LH→LH, and RH→RH arcs show. A switch from RH to LH and vice versa is rarely done by either speaker.

Handshape is determined by consulting a lexicon where legal handshapes for each gesture are specified (e.g. pointing can be done with the index finger or the open hand). Handshape selection now follows the rule of economy: if the handshape of the previous gesture is a legal handshape for the current one, then keep it. Otherwise change handshape to a suitable one.

After this stage of generation we have a sequence of gesture frames where lexeme, handedness and handshape have been specified.

5.2.3 *Creating Gesture Units.* The gesture hierarchy of phases, phrases and units is hypothesized to correspond to levels of speech phrase organisation. For instance, Kendon [Kendon 2004; 1980] suggested a correlation between *intonation units* and g-units. Such concepts go back to the hypothesis that speech and gesture originate from a single source, called *growth point* [McNeill 1992] or *idea unit* [Kendon 1980]. In our algorithm we try to approximate these concepts. We produce g-units by gradually merging gestures according to certain criteria. First, we take the first and the last gesture within a discourse segment and merge them with all in-between gestures to form a unit, i.e. adding a

g-unit edge to the graph. Then, we cluster neighboring g-units by merging all units whose distance in seconds is below a threshold θ_{unit}. We found $\theta_{unit} = 1.5$ seconds to be a good value. The threshold could be made speaker-dependent: a high value produces large units with many gestures, a low value produces more isolated gestures.

5.2.4 *Planning a Gesture Unit.* For determining phase structure, positions and timing, we string together suitable samples from the GestureDB and let emerging constraints guide the determination of phases.

The phase structure (prep, stroke, hold etc.) of a gesture G_i depends on the temporal constraints of neighbouring gestures G_{i-1} and G_{i+1}. If there is enough space up front to perform a preparatory motion ($> .5$ s) the gesture is assigned a preparation phase (which is always the case for the first gesture in a unit). If a gesture has a preparation, the positions of the gesture are unconstrained so we select a random sample of the respective lexeme from GestureDB. If a gesture has no preparation we find a sample whose start location best matches the end location of G_{i-1}. The chosen sample is used to specify positions, trajectory and type (s-gesture or h-gesture). To create multi-strokes for an s-gesture we consult the average number of multi-strokes, $\mu_{strokes}$, for gesture G_i. If $\mu_{strokes}$ exceeds a threshold we generate a random number of after-strokes using the mean value and standard deviation. If there is not enough space between the gestures and $\mu_{strokes}$ exceeds a yet higher threshold, then gesture G_{i+1} is either moved back in time, where the speech-gesture offset's standard deviation is an upper bound on how far it can be moved, or eliminated in favor of G_i's multi-strokes. For all other cases where there is space between G_i and G_{i+1} we generate a hold between them.

Now the main stroke of G_i can be precisely timed with speech by aligning the end time of the stroke with the end time of the gesture's arc in the graph, which corresponds to the end time of the word(s) that triggered this gesture. From this time point we subtract the speaker's average offset ΔT_{end} for this lexeme. We hypothesized that it is more important that gesture and speech *end* synchronously than that they start synchronously. To compute the timing of after-strokes we align all after-stroke end times with word end times enforcing a minimum duration for each after-stroke. This proved to be an efficient way of achieving after-stroke synchronization. For hold gestures we use a similar method but align *start* times of word and hold. Looking at the resulting animations with a virtual character we found that although the timing was similar to the original speakers' timing the gestures always seemed a little too late. Human speakers can supposedly vary the timing of their gestures with great flexibility because movements of the whole body, and especially the face, all contribute to gesture-speech synchronization. Since our virtual character has a comparatively limited expressiveness we make our gesture timing more conservative by subtracting a general offset of 0.3 seconds for main strokes and 0.12 seconds for after-strokes.

Using this algorithm we generate the following types of gestures: stroke, prep-stroke, prep-hold, prep-stroke-hold and stroke-hold, where all the strokes can be multiple strokes. As the g-unit's last gesture must by definition have a retract phase we have to determine to which rest position the hands return to. Observing the video material we determined three different rest positions: hands "at side", "in pockets", and "clasped". While the retraction after a unit could be modeled probabilistically we resorted to simple rules that work on the temporal distance to the following g-unit: if small, retract to "clasped"; if medium distance retract to "at side"; if far retract to "in pockets".

5.3 Body Movement and Gesture Script

Head and body rotations were generated to make the character look more alive. The generation is rule-driven rather than data-driven. The rules can fire on any kind of arc in the graph, in combination with keywords that occur in the range of the arc. For instance, we could write a rule that looks for an utterance arc U that contains the keyword "you" or "yours". If the rules fire, a rotation arc with a target direction (left/right) is added to the graph at the same position as U, meaning: the body rotates in the given direction when the arcs starts and rotates back to the default frontal position when the arc ends. This was inspired by findings that changes in body postures occur most often at boundaries of discourse units ([Cassell et al. 2001] and [Scheflen 1964]).

Body rotation rules were crafted from our observations of the subjects JL and MR. JL's monologue routine follows a clear pattern. It consists of a series of jokes, each taking about 10 seconds. Before a joke he turns right (JLs viewpoint), probably to read the teleprompter, and before and after the punchline he often turns left to address the band-leader. These special monologue units (joke opening, punchline and others) were specifically annotated in the JL data. Since MR is participating in a discussion, he rather turns to the person he addresses and stays there for a while. For MR, we use the utterance segment and keywords like "you" and "your" to trigger body rotations toward an invisible addressee to the left or right of MR. For our demo, we used the MR rules for text input coming from neither the JL or MR domain.

Head rotations are generated based on the generated body rotations. Both subjects anticipate their body rotations by turning their head a little earlier. JL also follows a gaze pattern in trying to distribute his gaze uniformly across the audience. Therefore, our algorithm inserts anticipating head rotations before body rotations and fills in random head rotations in-between body rotations.

The final graph is written to a linearized gesture script containing the following data: head rotations, body rotations and gesture units which contain one or more gestures and have a retract position specified. For each gesture the script specifies: lexeme, handedness, handshape, type (e.g. prep+stroke+hold or prep+hold), stroke/hold start time, multi-stroke start times, overall duration, number of strokes, gesture start location, gesture end location (only for s-type).

The Appendix includes a sample snippet of a gesture script. It begins with a preamble that defines the head and body rotations for the entire sequence. This is followed by a sequence of gesture units which are produced from the linearized graph. Each gesture unit consists of a sequence of gestures which contain the data listed above. Multi-stroke timing information is also specified as part of a gesture. The retraction pose is specified at the end of each gesture unit.

6. ANIMATION

The role of the animation system is to take the gesture script as input and produce a final character animation sequence. It does this by augmenting the data provided by the gesture script, mapping this completed set of data to a form that can be animated, and then producing an either kinematic or dynamically simulated animation.

The animation system used is an extended version of the one described in Neff and Fiume [2005], which is built on top of the DANCE framework [Shapiro et al. 2005]. Significant additions to the system include the use of offset layers and a set of augmentation processes that produce detailed animation specifications from the gesture script. The focus of this section will be on the new aspects of the system and how they are applied to the gesture animation task. The reader is referred to [Neff and Fiume 2005] and [Neff 2005] for other details on the system.

The system also generates facial animation for lip sync [Cohen and Massaro 1993] that takes into account coarticulation (i.e. the influence of surrounding speech segments on the vocal tract shape of a phoneme). Additional speech-related facial movement such as eyebrow raises on stressed phonemes is added based on universal rules [Albrecht et al. 2002]. Modeled variations between speakers are at the moment restricted to amplitude and frequency of eyebrow movement (very pronounced in JL, less salient in MR), but could easily be extended to include, for example, speaker dependent facial expressions for questions.

6.1 Underlying Representation

The prep-stroke-hold structure of gestures maps naturally to animation keyframes. Our underlying movement representation, therefore, is analogous to a keyframe system. Every DOF in the character's body has its own track, partial body poses are stored at particular points in time and transition functions (cubic Hermite curves embedded in space and time) control interpolation between these poses.

Our representation extends a traditional keyframe system in two ways. First, we support *offset layers* and second, we support non-DOF tracks that can be used to adjust real time processes. For any DOF, the desired value is determined by summing the main track with the data on any associated offset layers. While offset layers are a familiar tool for making low-frequency edits to motion capture data that preserve the high frequencies of the motion [Witkin and Popovic 1995], we employ them differently. We use them to add high frequency detail to our motion and also to layer different motion specifications together. The other novel type of track does not contain DOF values, but is used to control more complicated realtime processes in the system. For example, some parts of the body are controlled by real-time processes for gaze-tracking and balance control. The desired constraints for these processes are specified on these separate tracks in the underlying representation using the same key and interpolation function primitives as with other tracks, which supports continuous variation.

6.2 Data Augmentation

The gesture model needs sufficient control to correctly align gestures with speech and to reflect the key idiosyncrasies of a speaker's gesturing style. Gesture data is divided between the gesture model and animation engine in an effort to strike a balance between (A) the need for the gesture model to control the motion, (B) the desire to minimize the work required to annotate video for building the gesture model, and (C) the desire to allow the animation engine, which contains the relevant domain knowledge, to control the low-level aspects of motion production. The gesture script (Section 5.3) presents a minimal description of the required movements that captures the key definitional aspects that must be controlled by the gesture model. The animation system must augment this sparse representation, filling in more detailed data and adding important nuance. The process is one of refinement, continually adding more detail to improve the gesture rendering. Such an approach also allows for workload management as the animation can be generated after minimal augmentation, but adding more data to the animation lexicon will improve the quality of the animation.

The animation system performs a range of operations during data augmentation. It will:

— complete timing information

— deal with spatial conflicts due to the sparcity of spatial sampling

— add necessary definitional information for different gesture types

— add character specific variation.

These items will be detailed below. Character specific data includes global character properties [Neff and Fiume 2005], such as a default posture or tendency to start movements quickly, as well as variations on how a particular gesture is performed. Two global character properties were modeled for our subjects: default postures and a slight forward succession to the movements. Significant use of two techniques is made to augment the initial motion framework: *keyframe infilling* and the addition of *micro-keys*. Micro-keys are parameters that define partial body poses and can be layered on top of existing keyframes. Keyframe infilling is a process by which new keyframes are generated at locations in between the existing keyframes. These can be micro-keys (partial specifications) as well.

6.2.1 *Completion of Timing Data.* The gesture script specifies end times and durations for strokes as well as hold durations and start times for body rotations. The rest of the required timing data is determined by the animation planner, which can complete the data and also adjust for dynamic effects. The start time for prep movements is defined as:

$$prepStart = max(strokeStartTime - defaultPrepTime,\ lastStrokeEndTime) \tag{7}$$

where $defaultPrepTime$ is currently 0.4 s for preps within a gesture unit and 0.8 s for preps following a rest pose as these will have a longer distance to travel. A similar rule is used to determine the duration of the transition to rest poses where the time between the end of the last hold phase and the next stroke must accommodate both the transition to the rest pose and the subsequent prep phase. If there is enough time, 0.8 s is allowed for each. Otherwise, the time is split between the two movements, which is normally sufficient. Head movements are given a duration uniformly distributed between 0.2 and 0.3 s or until the start time of the next head movement if it is sooner. Body rotations are given a duration between 0.5 and 0.6 s.

6.2.2 *Spatial Augmentation.* To ease the annotation task, a relatively coarse spatial discretization is used to record gesture locations (Table III). This can lead to two problems: hand collisions and lost information for small movements. When generating animation, the system will place the hands at the middle of the spatial buckets corresponding to their discrete location tags from the annotation, which is generally sufficient, but can lead to conflicts. A small number of gestures, such as a two-handed wipe (Figure 3), feature the hands crossing over each other. When both hands cross, they may be annotated with the same discrete tags and a small two handed separation distance that will cause them to be placed in the same location when the data is used for animation. A slightly more common occurrence is for subsequent gestures to be given the same spatial location when there should actually be a small movement between them (even though both might still be in the same spatial bucket). Both of these cases are automatically detected and offsets are applied to the hands to correct them. A vertical offset is used to separate overlapping hands and a downbeat is applied to sequential gestures with identical locations.

6.2.3 *Additional Gesture Data.* In order to produce the final animation specification, the gesture description provided in the gesture script must be augmented with the additional data from the animation lexicon (Section 4.1.3). For each lexeme in the gesture script, the corresponding data is retrieved from the animation lexicon and added to the animation specification. Recall from Section 4.1.3 that this information includes palm orientation, posture changes and multi-stroke data. Most of this data is added to the animation specification using micro-keys - augmentations to existing keyframes at the stroke boundaries to specify specific DOF values such as wrist orientation, spinal bend, etc. Some changes such as warps to the transition curves are controlled using edits as described in [Neff and Fiume 2005]. Maintaining separate animation lexicons for each character allows character idiosyncracies to be modeled, such as specific posture changes for a given gesture.

The system automatically varies collar bone angles based on the gesture height. The form of after-strokes, which follow the main stroke, is also defined in the animation lexicon and added as part of the augmentation process. Both of these issues will be discussed below, in Sections 6.3 and 6.5 respectively.

6.3 Pose Calculation

The system uses a combination of keyframe and continuous motion generation techniques: discrete pose calculation with interpolation is used for gesture generation, and continuous IK for balance control and gaze tracking. During a preprocessing phase, the animation system maps the motion specification into the underlying representation that will be used to generate the motion. The poses for the start and end of each gesture phrase are calculated and stored in the DOF tracks. Body twists and certain posture changes are stored on the offset tracks. Curves specifying the desired gaze direction and balance point are stored on separate tracks. At each time step during playback, realtime processes will use the IK routines to update the lower body DOFs to meet the balance constraint. A second process will have the character look at the specified target by solving for the axial rotation of the head and neck, and the tilt of the head. A gain factor, set to .5, controls how far the head moves between staring straight ahead and staring at the look-at target. The remaining DOFs of the body will be determined from the other tracks.

Constraint Height	*Base Offset (deg)*
above-head	-5
head	-3
shoulder	-1
chest	2
abdomen	3
belt	5
below-belt	7

Table VIII. Base offset angles applied to the collar bones. These are multiplied by a character specific scale factor.

6.3.1 *IK and Pose Determination.* Rather than a monolithic IK system that solves for an entire posture, we use a set of simple, analytic IK routines that are each responsible for a portion of the body. Lower body movement, including knee bends, pelvic rotation and balance control, is based on an analytic lower body IK routine and feedback based balance adjuster. To avoid stability problems in dynamic simulation, springs are used to hold the character's feet in position and prevent him from falling. Analytic routines are used for arm positioning, and aesthetic constraints are blended together to determine the torso pose. The pose solver is described in [Neff and Fiume 2006], except that we disable the optimization routine described there as we are not using spatial constraints to deform the character's posture in this work. We also augment that system with automatic collar bone adjustments and add local IK routines to achieve hand and wrist orientation constraints.

The process for calculating each pose in a gesture phrase is as follows:

(1) Calculate the posture of the spine and collar bones. This requires blending constraints for the character's default posture with posture constraints from the animation lexicon. Automatic collar bone offsets are also introduced here.

(2) The arms are positioned to meet the wrist constraints specified in the gesture description.

(3) The arm swivel angle is rotated to meet the inclination constraint.

(4) Constraints on palm orientation are solved.

The automatic collar bone adjustment offsets the shoulders up or down based on the location of the reach constraint. The base offset for each of the constraint heights is shown in Table VIII. These base offsets are multiplied by a character specific gain value, which is 1 by default. Collar bone adjustment is important for increasing the naturalness of arm movement. This adjustment reflects the biological construction of the shoulder as the upper arm and clavicles are not independent joints (for a discussion of models of the shoulder complex, see [Badler et al. 1992].)

The gesture targets used when positioning the arms are defined to be relative to the current orientation of the body (cf. Table III), but are not defined in the frame of a particular joint. For example, a shoulder height target will remain at shoulder height as the character hunches over. Each height target, such as "shoulder", "chest" or "abdomen", has a defined height within a particular limb in the skeleton, which can be used to determine a world space height constraint. The radial distance from the character's centre and the distance in front of the character's body both lie on the world space horizontal plane with the given height value. The distances from the character's body ("touch", "close" etc.) are defined relative to the body part the gesture is in front of. The radial inclination, however, is defined relative to the chest. This means that a "front" radial inclination will be in front of the chest even if it is at belt height and the torso is rotated. These definitions perform well with the annotator's expectations of the markup scheme.

6.3.2 *Specifying Body Rotations.* Body rotation is discretized into three directions (left, right, front) and gaze direction is discretized into five locations (left, left-front, front, right-front, right). The exact values of these locations are

(a) Straight Trajectory (b) Curved Trajectory

Fig. 10. Comparison of a straight and curved trajectory for the same *cup* gesture.

specified for each character based on the video corpus, with JL having larger rotations. Body rotations are accomplished by a combination of a pelvic rotation with opposing knee bend and a rotation of the abdomen and chest spinal joints. Each of these rotations are offset in time by ten percent of the rotation's duration to give a more natural flow to the motion. These rotations are specified on offset layers and blend with other posture deformations.

6.4 Keyframe Refinement

In addition to the layering of micro-keys on top of existing keys, as described in Section 6.2.3, some gesture attributes require the creation of additional keys, as detailed here.

6.4.1 *Path-in-Space.* Whether a movement follows a straight or curved path in space is an important expressive property. In our previous work, we did not model this property [Neff and Fiume 2005]. Chi *et al.* [2000] in the EMOTE model represent it by varying the trajectory of the arm end-effector and also provide three different interpolation spaces: interpolating joint angles, end effector position or elbow position. Similarly, Kopp and Wachsmuth [004c] use guiding strokes in space that allow the curvature of a motion to be controlled. Unlike the previous approaches, we achieve satisfying curved motions by working in joint space using offset curves, rather than working in world space. This approach is simple and avoids the need to determine and orient world space trajectories to try to provide a natural path for the motion.

There are two main types of curvature we need for our gesture lexicon: point-to-point curvature, where a single stroke follows a curved path, and continuous circular movements for gestures like progressives. The latter case will be discussed below. By default, movements between two points in our system will produce a basically straight path[12]. A curve can be added to the motion by introducing an offset perpendicular to the path of the movement that starts at zero, peaks near the middle of the movement, and returns to zero. We normally apply these offset keys to the elbow and larger amplitude offsets will produce a higher curvature motion. For example, a "cup" movement that has the hand up and a largely horizontal trajectory, can be curved by adding an offset to the elbow bend as shown in Figure 10.

6.4.2 *Progressives.* A *progressive* is a cyclical movement in which the forearm and hand are moved in a circular loop in front of the chest, first coming towards the body and up, then out from the body and down (Figure 3). There may also be a translational component as the centre of rotation is moved through space. *Regressives* consist of the same motion rotating in the opposite direction. A progressive (or regressive) is specified by indicating in the gesture script

[12]Within the limits of basic quaternion interpolation of the shoulder, which can introduce some warping to the path.

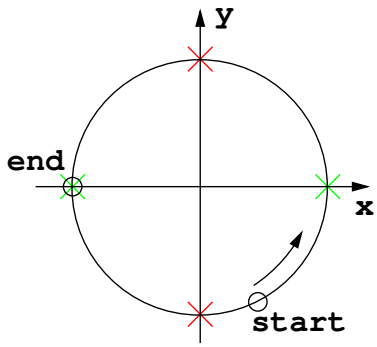

Fig. 11. The wrist will trace a circular path during a progressive (ignoring the translatory motion). Keyframes are located at the coloured crosses. The movement travels between the two small circles, along the arc of the large circle.

where in space it starts and stops, the number of rotations, and which hand(s) are used. Multiple partial keyframes (keyframe infilling) are then used to generate the motion. To our knowledge, previous embodied conversational agent systems have not modeled gestures with such complicated spatial trajectories.

The basic idea of how the curvature of the motion is created can be understood by considering a 2-DOF pendulum. Applying ninety degree out of phase sine waves to the pendulum will cause its end to trace out a circular path. We apply this idea to the character's forearm and also apply similar rotations to the wrist. Instead of using sinusoidal interpolation functions, a key is placed at every 180 degrees in the sinusoid and connected with ease-in ease-out curves, effectively approximating a sinusoid. Since there is also translational movement during the progressive, the overall movement is decomposed into two components: one that determines the rotational movement with the approximated sinusoids and one that specifies the translation in space. The overall process is as follows:

```
(1)  Determine the amplitude of the circular movement
(2)  Determine the time that each infilled keyframe must occur at

     For each infilled keyframe
(3)      Determine the x rotation and corresponding hand rotation
                          - OR -
(4)      Determine the y rotation and corresponding hand rotation

(5)      Add the keyframe to the system

(6)  Add an offset curve to account for elbow translation
```

It is important that the scale of the rotation be proportional to the range of the translational movement as one loop of tight rotation stretched over a long distance tends not to have the look of a progressive. Consider the circle in Figure 11 as representing only the rotational component of the movement. The circle has a particular size, determined by the magnitude of the input sine waves, so covers a certain area in space. The challenge is to relate this spatial size created by the rotation to the overall spatial coverage determined by the start and end constraints. More intuitively, it appears from our samples that there exists a correlation between larger rotations and larger translational components during a progressive gesture and vice versa. To relate the two quantities, we determine the amount of rotation (i.e. the magnitude of the sine wave) that would generate a circle just large enough to span the two end constraints. A new quantity is defined, *maximal rotational amplitude*, a, which is the amplitude that must be applied to the forearm to

achieve this size of rotational circle. This maximal rotation is then divided by the number of cycles of the progressive, n, and a character weighting factor, m, to determine the magnitude of rotation, α, for each loop of the progressive (step 1):

$$\alpha = \max(MIN\_AMPLITUDE, m\frac{a}{n}) \tag{8}$$

where MIN_AMPLITUDE is a minimum rotation that is provided to still create a progressive if the end constraints are the same. The character specific multiplier, m, is 1 by default.

In the final representation, the keyframes for the x DOF and corresponding hand movement will be placed at the green extrema in Figure 11 and the y keyframes at the red extrema. By default, progressives start at the position marked "start", $5/6\pi$ for counter-clockwise rotation, and end at the location marked "end", $-1/2\pi$ for counter-clockwise rotation. The time of the keyframes is determined by considering the total duration of the movement and the number of cycles of the progressive and then calculating the appropriate keyframe spacing (step 2).

Steps 3 and 4 determine the desired angles at each keyframe (note that each keyframe has data for either x or y, but not both). Different angle representations require different approaches. With Euler angles, as we use at the elbow, the values are simply $\pm\alpha$ as appropriate. The shoulders are more complicated as they are represented with quaternions and hence do not have a separate component corresponding to the axial upper arm rotation. In determining these keys, we combine the rotational component of the progressive and the overall translatory aspect of the movement that will be achieved by x rotation[13]. We first update the orientation of the upper arm that is used to move the hand between the end constraints:

$$q_i = slerp(q_0, q_1, p) \tag{9}$$

where q_0 is the quaternion satisfying the initial constraint, q_1 for the final constraint, q_i is the infill quaternion we are calculating and $p \in [0,1]$ is a progress variable indicating how far we are between the beginning and end of the progressive. This calculation corresponds to the translatory portion of the progressive. The rotational component is achieved by rotating q_i around a vector aligned with the axis of the upper arm by $\pm\alpha$ as appropriate.

In step 5, these keyframes are added to the underlying representation. The translatory component of the elbow movement has not been accounted for yet. This is done by adding a linear offset curve for this DOF (y) that spans the translation (step 6).

An example of a fairly large progressive is shown in Figure 12. Notice that there is a diagonal translatory component to the motion of the gesture as well as the core circular movement of the progressive. A trace of the hand's path shows a circle stretched in time.

6.5 After-Strokes

A multi-stroke consists of the main stroke phase followed by a number of smaller repetitions we call *after-strokes*. Consider a "dismiss" in which the character raises his arms and hands (prep), and then flings his hands down, letting the wrists go limp (stroke), followed by two small repetitions in which the character raises his hands slightly and tilts his palms back up towards the audience before moving the hands back down and dropping the wrists once again. These repetitions are *after-strokes*.

There are at least two categories of after-strokes. The first consists of essentially continuous, rhythmic hand waving at the end of the stroke. MR frequently uses such gestures. The second has the same prep-stroke-hold structure as

[13]These two components correspond to a sinusoid in x and a linear ramp in x that are summed to provide the final value.

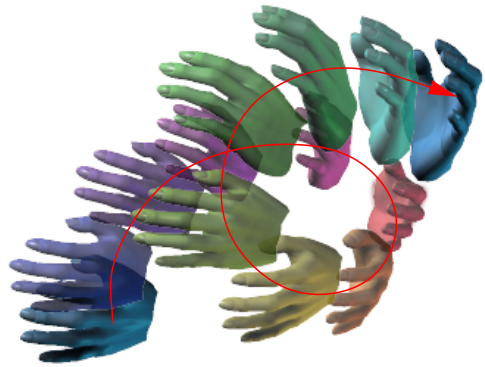

Fig. 12. The path of a hand during a large progressive with a diagonal translatory component. The colour gradation is used to show the progress of time.

the main gesture as in the "dismiss" example above. The hold period in these after-strokes is particularly important. When the dismiss is repeated without a hold phase after the stroke, it disintegrates (disappears) into hand waving. The hand appears to bounce up from the end of the motion and the definitional downward aspect of the movement is lost without the necessary pause at the stroke end. It should be noted that these "prep" and "hold" phases mentioned here are not explicitly modeled in the grammar rules of Section 3, but are contained in the notion of an after-stroke.

Most data associated with after-strokes is local as they are normally rapid movements that are confined to the wrists and forearms. The animation lexicon (Section 4.1.3) accepts any subset of the following data to define an after-stroke pose: offset to vertical or horizontal wrist positions, forearm rotation, wrist rotation (2 possible DOFs) and an offset to the elbow bend. This data is defined separately for each of the two movements making up an after-stroke (nominally, prep and stroke). It is only defined for one arm and mirrored to the other.

After-strokes are created by keyframe infilling. Copies of the stroke keyframe from the end of the main stroke are made and used as the basis for both the prep and stroke poses of each after-stroke. This is done as we wish to add small, local variation to the end position of the main stroke, rather than copying the often larger spatial movement of the main stroke. Additional attributes are then added to these copies based on the data in the animation lexicon.

The timing of these keyframes is illustrated in Figure 13 and calculated as follows. The duration available for the complete after-stroke is $d = e_i - e_{(i-1)}$ and must contain the prep and stroke associated with the after-stroke, plus the hold from the previous stroke. This is because the time constraints e_i and $e_{(i-1)}$ from the generation algorithm specify the end time of strokes. The animation lexicon defines the percentage of time that should be spent on each of the prep, stroke and hold phases. In the case of holds, the previous duration, $e_{i-1} - e_{(i-2)}$, is used to calculate the hold time as the hold corresponds to the stroke from that duration. If there is more time in d than is required by the hold, the hold is expanded to fill the available time. If there is less time, the actual hold is set to be the average of the calculated time and the time available. The prep and stroke phases then have their duration decreased to fit each phase within d.

After-stroke durations may vary widely and we wish to reuse a single definition for each after-stroke of a given lexeme. To avoid unnatural movements when after-strokes are very short, limits are placed on the average spatial and angular velocity of after-strokes for kinematic animation (these limits are unnecessary in the dynamic case). If these limits will be exceeded, the spatial or angular range is reduced so that the average velocity is not exceeded.

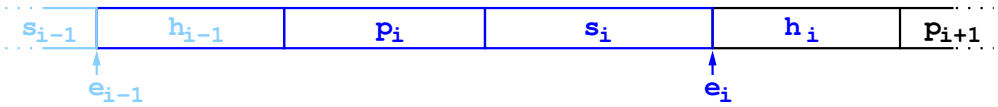

Fig. 13. The end times e_{i-1} and e_i are constraints specified by the gesture generation module. The remaining timing of the after-strokes must be calculated to fit into these intervals. Here h indicates a hold, p a prep and s a stroke.

6.6 Physical Simulation

Physical simulation can improve the realism of the resulting gestural animation in several ways. First, it will smooth the motion in a natural way. Second, there is very little basis in the collected data for providing small torso deformations that are often caused by arm movement during gesturing. Simulation allows the transfer of force from rapid arm movements into the torso which can cause these deformations and improve the realism of the motion. Third, the damping in the model will limit the speed of any movements with unreasonably high velocities. Finally, simulation can add small end-effects to the motion, such as pendular arm sway when a character brings his arms to his side or passive movements of the fingers.

When computing physically simulated animation, we use a controller based approach whereby an actuator is placed at each DOF which calculates the torque required to move the character towards the desired configuration. We use an antagonistic formulation of proportional-derivative control, following [Neff and Fiume 2002]. The control law is written as

$$\tau = k_L(\theta_L - \theta) + k_H(\theta_H - \theta) - k_d\dot{\theta}, \tag{10}$$

where τ is the torque generated, θ is the current angle of the DOF and $\dot{\theta}$ is its current velocity. θ_L and θ_H are the low (L) and high (H) spring set points which serve as joint limits, k_L and k_H are the corresponding spring gains, and k_d is the gain on the damping term. The tension T or stiffness of the joint is taken as the sum of the two spring gains: $T = k_L + k_H$.

Consider the stroke phase of a "cup" gesture (Figure 3). When this phase begins, the spring gains for the current pose and the gains needed for the desired pose at the end of the stroke are computed for each DOF used in the motion. The movement is generated at each time step by interpolating between these gain values, determining the torque that the current gain values will generate given the current state of the character, and then using the equations of motion to determine accelerations that are twice integrated to update the position of the character. The torques generated by gravity at the start and end pose are calculated using the current state of the character and an estimate of the end state. The gain values are computed to compensate for these torques (this process is represented by the function C below). This allows joint tension to be varied during a movement while still ensuring joint positioning that is accurate, at least at steady state.

In our system, offset tracks are summed with the main track to produce final control values. In kinematic simulation, each track contains angle data that can be directly added. In physical simulation, we add the gains. Consequently, gains must be calculated across tracks such that the they will add to the correct values to produce the desired angles. A function C can be defined which takes a desired angle θ and computes the opposing spring gains (i.e. $(k_H, k_L) = C(\theta)$) that will balance the forces acting on the limb such that the equilibrium angle of the limb is θ. This function must perform gravity compensation. Given C, the rules summarized in the table below can be used to compute gains on each track:

	Main Track	*Offset Track*
Initial Gains	$(k_H, k_L) = C(\theta - \theta_{offset})$	$(k_H, k_L) = C(\theta) - C(\theta - \theta_{offset})$
End Gains	$(k_H, k_L) = C(\theta_{main})$	$(k_H, k_L) = C(\theta_{offset2} + \theta_{main}) - C(\theta_{main})$

These rules follow from Equation 10. θ is the current angle for the DOF at the start of the movement and θ_{offset} is the current offset angle. For the end point gains, θ_{main} is the desired value on the main track at the end of the transition and $\theta_{offset2}$ is the desired offset value at the end of the transition. These values are estimated based on the transition curves associated with the DOF and offset tracks. The tension is kept the same for each component during a transition, but the start and end times of the main and offset curves do not need to be the same.

Aside from the balance problem which we mitigate by using springs to hold the feet in position, one of the main difficulties encountered with controller based simulation is setting the gain values appropriately to generate the desired visual appearance. The inertia weighting technique presented by Zordan and Hodgins [2002] provides a good initial estimate for joint gains. We augmented this by an automatic sampling procedure that takes repetitions of a prototypical movement and computes gain and damping values that would yield a specified overshoot for a given DOF (e.g. a two degree overshoot in elbow angle at the end of the transition). This yields tables of tension and damping values that are useful for fine tuning the parameters when required. This tuning process is done once per character and then used to generate all animations. During retraction phases, we relax the character's hands. The gains used for this are calculated based on the approach described in [Neff and Seidel 2006].

The rapid, time synchronized movements in gestural animation are a challenge to model using a proportional derivative control approach due to the damping needed to stabilize the system. The actuators include damping, which is important for producing realistic motion. However, as we wish to operate at relatively low-tension levels that will enable the movement to be enhanced by many of the benefits of physical simulation listed above, the damping in the system will introduce lag. This is particularly significant when dealing with short duration, high velocity movements with precise timing. The lag causes two problems: first, it means that the movements will be slightly slower and so will be behind their desired time constraints. Second, if the kinematic trajectories are used as the basis of the PD-control, the extent of the movement will be reduced in many cases as the movement will not have time to reach the desired end-point before the control trajectory changes direction. We must compensate for both of these effects.

To ensure that simulated movements satisfy the script timing, we moved the start time of all poses earlier in dynamic simulation. Empirical tests showed -0.12 s to be an appropriate offset. With this offset, the initial time of transitions corresponded well to that seen in the kinematic motions. There are two potential ways to maintain the extent of the specified motion: the desired trajectory curves could have their extent increased, pulling the motion closer to the actual target, or the duration of the movements could be shortened and pauses inserted, allowing the actual movement to "catch-up" with the desired trajectory. For most cases, we use the latter approach. The pauses allow time for the motion to complete before a direction change begins. The update rule for the duration of strokes, d_s, is: $d_s = min(d_s, max(.15, d_s - .3))$ and the update rule for the duration of preps, d_p, is: $d_p = min(d_p, max(.1, d_s - .15))$ where the basic intuition is to reduce the duration of the motion to allow completion while still maintaining some minimal transition time. The offsets were determined on test motions, with a shorter offset being used for preparations as they normally have shorter base durations.

In the case of progressives, the continuous timing of the motion is particularly important, so we increase the extent of the transition curves rather than shortening the duration of the movement components. This is achieved by multiplying the angular span of the movement α by a factor of 1.6.

A variable time step Rosenbrock integrator [Press et al. 1992] is used to compute the motion using simulation code from SD/Fast [Hollars et al. 1994]. A 58 s MR sequence computed in 14 minutes and a JL generated sequence of the same length took 10.5 minutes on a 3 GHz Pentium 4.

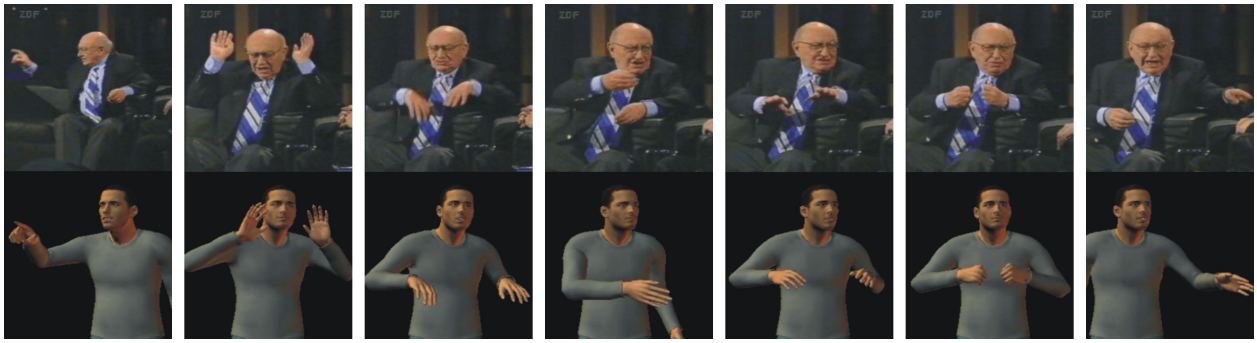

Fig. 14. Frames of MR gesturing from the video corpus and frames of the same movement from animations recreated from the corpus. Note: In the fourth pair, MR is making a RH gesture and his left arm is at rest.

7. RESULTS

A video was produced for this work that includes three pairs of example animations produced by our system for the two subjects. In the first example, we show for each subject a re-creation of a particular sequence of their video corpus. The gesture scripts for these sequences are created directly from the video annotation (Figure 2), and kinematic animation is used. These sequences validate both the fidelity of the annotation process and the ability of the animation system to generate the specified movement in the gesture script. A comparison for several gestures of a recreation of MR data to the original corpus is shown in Figure 14.

The second examples use the same audio tracks as the first, but generate new gestures based on each speaker's respective gesture profile. Since this input was also part of the training data we subtracted the statistical data for these particular samples from the profiles before generating the new gesture scripts. These sequences were dynamically simulated. Although the generated gestures generally differ from the original (or the recreation) the animations distinctly reflect the gesticulation patterns of the modeled individuals. The resulting animations present effective gesture timing, synchronized with the original audio, and gesture forms that are consistent with the modelled subjects. This offers validation for the generation model.

The final pair of examples show gesture sequences generated by each of the subject models and dynamically animated for a new passage of synthesized English text that is not contained in either video corpus. This demonstrates that our system can operate on novel text and is language-independent, since MR's gesture profile was built on German training data. The video also illustrates the role movement plays in creating the overall impression of an utterance. Even though the timing of the speech is unlike that of either subject, the resulting animations are characteristic of each speaker.

A side by side comparison of kinematic and dynamic animations reveals small differences in timing, but the overall synchronization remains intact. A comparison of the strengths of each approach to animation is included in Table IX. The strengths of the kinematic approach relate to computational cost and ease of use. The strengths of the dynamic approach, meanwhile, relate to the addition of subtle movement details that are lacking in the kinematic animation. Although subtle, we feel these effects help give the character a sense of "aliveness" that is less strong in the kinematic animation. This further validates the use of physical models and shows their relevance to synchronized gestural animation, where they have not previously been used. At the same time, as better tuning and control methods are developed, we postulate that the contribution of physical models to movement quality will become even more substantial.

A side by side video comparison of the two models used on the same text sequence nicely shows the style differences

Advantages of Kinematic Animation
Easier to precisely control timing.
Takes less time to compute, so is more suitable for real time applications.
No tuning is required.

Advantages of Dynamic Animation
Small oscillations/overshoot at the end of movements reflect momentum. Fast movements will cause small oscillation in the final position at the end of the movement that will not occur for more controlled, slow movements and are absent in the kinematic animation. This effect is most obvious when a character drops his arm to his side, producing pendular motion, but also occurs at the end of strokes and other gesture phases.
Gravity will vary the relaxed hand-shape based on the orientation of the palm whereas kinematic hands remain stiff.
Arm movements cause some force transference into the torso that generates slight deformations to the torso adding realism.
Secondary movement is included. For instance, when a character rotates, there is a slight swinging of the arms that is lacking in the kinematic version.
Better modeling of interrupted movement. For example, in one sequence JL drops his hand towards his side, but then begins another movement just before his hand reaches his side. The dynamic model nicely captures the transition from passive to actively controlled movement.

Table IX. Comparison of strengths of kinematic and dynamic approaches to animation generation.

Fig. 15. A matter of style: Different gestures were generated for contemporaneous frames of the "Star Wars" animations for JL (top row) and MR (bottom row).

between the two models (JL vs. MR). Frames from this are shown in Figure 15. Worth noting, not only does the system produce different gestures for each speaker, it also generates very different, yet still effective, timing patterns. For instance, the last pair of frames in the figure show a case where a gesture is generated for the JL model but not the MR model.

Overall, the animations show a high variation in gesture shape, good synchronization with speech and a nice overall flow of movement. High variation stems from using positional data from the GestureDB and from creating multiple strokes. The good synchronization validates our algorithms for aligning main stroke and after-strokes, using Steedman's concept of *focus* as an important gesture placement indicator. Finally, the overall flow is due to our introduction of the *gesture unit* as an organizational higher-level entity.

7.1 Validation

An evaluation study of our system was conducted that shows that the gestures produced by the system are recognizable as having the style of the specific performer modeled[14]. For this study, 26 independent reviewers were recruited, aged 24 to 46; 6 female and 20 male, all non-expert in the field of gesture modeling and/or animation. In a learning phase they were shown video clips of the original performers, JL and MR. Two clips of each performer were used, about 5 minutes in total. These clips were outside the training corpus used for our statistical models. The order of the clips was varied across subjects to avoid order effects.

In the first test (Test 1) we showed them one video clip of generated gestures and asked "Whose gesturing style is imitated in the first animation?". The JL model was used for half of the subjects and the MR model for the other half. The animations for the novel text were used in the experiment (these are the last two clips discussed above) which are obviously outside our training corpus. Afterwards, in the second test (Test 2) we showed a side-by-side clip of both variants of the novel text clips (modeled on JL + modeled on MR) and asked subjects to "Please indicate which clip is animated more in the style of Jay Leno (JL, American) and which more in the style of Marcel Reich-Ranicki (MRR, German).".

The result of Test 1 was that subjects selected the correct original performer 69 % of the time, which is significantly above chance ($t(25) = 2.083; p < .05$). The result of Test 2 was that subjects correctly assigned the original performer to the side-by-side characters in 88 % of the cases which is also significantly above chance ($t(25) = 6.019; p < .001$). In Test 1, there was no significant difference in recognizing JL compared to recognizing MR ($t(24) = .5708; p = .57$). For both tests, there was no noticeable dependency of subject performance on gender, age, familiarity with English / German or with either performer.

We did not formally evaluate how subjects identified speakers, but based on post survey discussions, it appears that different subjects used different clues. Some relied on a few distinctive gestures that they thought were typical for the performer, while others paid more attention to rest pose and others focused on their overall impression of the sequences.

The evaluation clearly shows that the produced animation reflects the style of a specific performer. This worked equally well for both performers. When putting the animations side by side, the discrimination task is even easier, as reflected by the higher scores. It should be noted that the selected clips were generated on synthetic text for which we did not model the timing and speaking pattern of either speaker. This makes the recognition task harder as these important aspects of personal style were absent in the stimuli, providing less information than would be present in a clip of either speaker.

7.2 Expertise Required to Use the System

Our approach requires manual work that can be performed by non-experts with some training. For a more specific estimate we have to distinguish between the labour-intensive *analysis phase* and the *runtime system*.

In the *analysis phase*, for the coding of words, theme/rheme and discourse segments a linguistic background is helpful. The coding of gesture phases and their spatial properties requires no special prior knowledge. For encoding the lexical affiliate, some knowledge of the gesture literature, especially on gesture inventories and abstract/metaphoric gestures [McNeill 1992], must be acquired in training. Creating the animation lexicon requires good observation skills as the annotation is based on categorizing what the performer is doing in still images. The annotator must also be familiar

[14]This experiment was done using a slightly earlier version of the system. The quality of the animations has subsequently been improved.

with the meaning of the parameters in the system so that he can represent what he observes. In all, we estimate a training period of 1–2 weeks for the analysis phase.

In the *runtime system* the manual labour consists of preparing the input data, i.e. adding theme, rheme, focus and utterance segmentation, and extending the semantic tag look-up table by adding unknown words that fit into one of the categories. For both tasks a linguistic background is helpful. We estimate a training period of less than a week. However, note that automated approaches for reconstructing these data from plain text exist, e.g. in BEAT [Cassell et al. 2001], and could be added to our (runtime) system to make it run without manual work.

7.3 Data-specificity of the Approach

While our approach strives to model individual style in gesture behaviour, it is at the same time quite general. The animation can handle a very wide range of possible gestures and the style-specific data is encapsulated in character-specific gesture profiles, separate from a general gesture generation algorithm. In this section, we discuss the limits of the system in terms of domain, range of gestures and cultural dependency.

Our use of semantic tags make the approach independent of both language (German/English) and conversational domain. Looking at the table of semantic tags (Table VII) one can imagine that they are applicable to many domains. In fact, the data for our two speakers comes from quite different domains (book review vs. comedy). In our demo, we explicitly chose input text from a totally new domain, the prologue from the Star Wars episode IV movie, to show that our approach transfers to other domains.

While the semantic tags themselves are quite domain-independent the concrete look-up table must be extended for each new text. Therefore, the lookup table should be replaced by an automated approach. The animation lexicon must be extended each time a speaker is added; if the hypothesis is true that all people from the same cultural group draw from a single repertoire of gestures (excluding iconics), then this work becomes less with a growing set of profiles. Even for our two speakers, we found a considerable overlap in gesture lexemes. However, the lexicon of gestures may be culture-specific. This means that adding a person from a culture different to the corpus may entail more work on the animation lexicon as opposed to adding people from the same group.

As discussed previously, the parameters in the animation lexicon most related to the particular individual performing the gesture are those related to posture changes. After-strokes might also show individual varation, but there is not enough data to verify this. It appears that palm orientation is defined more by the lexeme with limited variation across individuals.

Although the range of produced gestures is quite large, our approach misses out on iconic gestures that illustrate complex spatial content (e.g. the trajectories of two colliding cars) or make deictic reference to present objects (e.g. pointing to a moving object). However, such gestures could be added on top of our approach by a “deep” generation engine.

8. CONCLUSION AND DISCUSSION

This work presents a system for generating believable gesture animations for novel text that reflect the gesturing style of particular individuals. It moves beyond previous approaches by creating a statistical model of particular individuals; modeling gestures at a high level of detail; modeling complex gestures, including highly variable after-strokes and progressives; building gesture units that flow well and synchronize effectively with speech; and using physical simulation to enhance the final animation. Gesture units are a particularly effective construct. We generate stretches of gestures and infer timing parameters from the interdependence of the gestures contained in one unit. This makes gesture flow much more natural as the gestures are connected by holds or directly succeed each other while

positional parameters are fitted depending on the preceding gesture. We found the alignment of end times between gesture strokes and speech correlates performed well. Finally, we exploit Steedman's concept of *focus* to synchronize gestures not with the directly related part of the utterance (lexical affiliate) but with the focus. The successful timing this produces breaks the myth common in the literature of a gesture having to occur slightly before its semantic correlate.

It is worth highlighting the effective division of data between the gesture generation process and the animation process. The gesture generation system contains the data necessary to both synchronize gestures with speech and to capture the spatial gesturing pattern of particular individuals. The animation system supports the reuse of labour through the animation lexicon, hides many details of animation production from the gesture generation process and effectively augments the generation data to produce convincing gestural animation.

Automatically creating animations of talking characters that reflect a specific subject's style and will satisfy a human observer is a very challenging task, and much work remains to be done. First, although the annotation scheme is simple, elegant and effective, the process is currently labour intensive. We see significant opportunity for automation in the process. Research has already been published on tracking JL's face and hands [Tan and Davis 2004], which can likely be extended for use in our workflow. Second, certain movements were avoided. For instance, handrub motions require a high quality collision model; iconic gestures need a deeper model of semantics. As well, the system should be extended to model non-hand based gestures such as head points and shoulder shrugs without accompanying hand movement. Better models for torso engagement while gesturing are also worthwhile. Another interesting avenue for future research is to directly model variations in expressivity. For instance, the amount of posture variation present in the performance of a lexeme is likely correlated with the amount of emphasis, excitement, anger or other intentional parameters related to how the subject is expressing the idea. Finally, we conjecture that better use of physical models can further improve the quality of the animations. People continuously modulate the tension in their bodies while moving, in a much more complex way than modeled here. Despite this, it is worth noting that this work demonstrates that physical simulation can add subtle details to the animation of gesturing characters. Using physical simulation for motion generation offers the potential to unify skeletal character animation with secondary effects like cloth modeling, all within the physical simulation realm. This will help ensure that character motions exert reasonable forces on these secondary models and offers the potential to create a continuous representation from limb movement, to muscle and skin deformation, to cloth and hair.

Acknowledgements

First and foremost, the authors would like to thank Jay Leno and Marcel Reich-Ranicky for providing such excellent samples of gesturing as part of their normal routine. The authors are also grateful to Jürgen Trouvain for the phoneme transcriptions, Ari Shapiro for support with version 3 of the DANCE framework, Nuance for providing us with a TTS system (Realspeak Solo, male, English voice) for our third example video and all the participants of the evaluation experiment.

The first author was a fellow at the MPI Informatik when much of the work described in this paper was completed. This work was partially funded by the German Ministry for Education and Research (BMBF) as part of the VirtualHuman project under grant 01 IMB 01A.

Appendix: Gesture Script Example

A section of a gesture script is included below. See Section 5.3 for a discussion of the gesture script.

```
BEGIN_BODY_ROTATIONS
  ROTATE_BODY right 10.073230361
  ...
END_BODY_ROTATIONS

BEGIN_HEAD_ROTATIONS
  ROTATE_HEAD right 9.673230361
  ROTATE_HEAD front 17.305170822
  ...
END_HEAD_ROTATIONS

#-------------------------------------------------------
BEGIN_G_UNIT
  BEGIN_GESTURE
    # start time = 8,704
    # Triggered by: "civil war" AGGRESSION 10,473 - 11,188
    lexeme=Fist    # from sample 32  (random)
    handedness=2H
    handshape=fist
    type=prep+stroke+hold
    # total stroke duration = 1,685
    stroke.trajectory=straight    # from sample 32
    hold.time.duration=0.4880431763966211
    stroke.time.duration=0.6936347179353073    # ran. offset 0,289
    stroke.time.end=9.897439842000004    # offset -1,291
    stroke.number=3
    mstroke.0.time.end=10.485929679
    mstroke.1.time.end=10.88846035
    stroke.position.start.2h_distance=0.6537408296950502
    stroke.position.start.height=shoulder
    stroke.position.start.distance=close
    stroke.position.start.radial=front
    stroke.position.start.inclination=normal
    stroke.position.end.2h_distance=0.8116517219742472
    stroke.position.end.height=chest
    stroke.position.end.distance=close
    stroke.position.end.radial=front
    stroke.position.end.inclination=normal
    # end time = 11,377
  END_GESTURE

  BEGIN_GESTURE
    # start time = 11,377
    # Triggered by: "galactic empire" TITLE 16,944 - 18,005 [init]
    # sync'd with Init words "rebel space"
    lexeme=Umbrella    # from sample 77  (random)
    handedness=LH
    handshape=open-rlx
    type=prep+stroke
    stroke.trajectory=straight    # from sample 77
    stroke.time.duration=0.2074158836033774    # ran. offset -0,013
```

```
    stroke.time.end=12.08391941    # offset -0,535
    stroke.number=1
    stroke.position.start.height=belly
    stroke.position.start.distance=normal
    stroke.position.start.radial=out
    stroke.position.start.inclination=normal
    stroke.position.end.height=belly
    stroke.position.end.distance=normal
    stroke.position.end.radial=out
    stroke.position.end.inclination=normal
    # end time = 12,084
  END_GESTURE

  ... # more gestures

  RETRACT_GESTURE pose=at-side
END_G_UNIT

... # more gesture units
```

REFERENCES

ALBRECHT, I., HABER, J., AND SEIDEL, H.-P. 2002. Automatic generation of non-verbal facial expressions from speech. In *Proc. Computer Graphics International 2002*. 283–293.

AXTELL, R. E. 1998. *Gestures - The Do's and Taboo's of Body Language Around the World.* John Wiley & Sons, Inc., New York. First published in 1991.

BADLER, N. I., PHILLIPS, C. B., AND WEBBER, B. L. 1992. *Simulating Humans: Computer Graphics, Animation and Control*. Oxford University Press.

BOERSMA, P. AND WEENINK, D. 2005. Praat: doing phonetics by computer (version 4.3.14) [computer program]. Retrieved from http://www.praat.org/.

CALBRIS, G. 1990. *Semiotics of French Gesture*. Indiana University Press, Bloomington, Indiana.

CASSELL, J., NAKANO, Y., BICKMORE, T., SIDNER, C., AND RICH, C. 2001. Non-verbal cues for discourse structure. In *Proc. Annual Meeting of the Association for Computational Linguistics 2001*. 106–115.

CASSELL, J., PELACHAUD, C., BADLER, N., STEEDMAN, M., ACHORN, B., BECKET, T., DOUVILLE, B., PREVOST, S., AND STONE, M. 1994. Animated conversation: Rule-based generation of facial expression, gesture and spoken intonation for multiple conversational agents. In *Proc. SIGGRAPH 1994*. 413–420.

CASSELL, J., VILHJÁLMSSON, H., AND BICKMORE, T. 2001. BEAT: The Behavior Expression Animation Toolkit. In *Proc. SIGGRAPH 2001*. 477–486.

CHI, D. M., COSTA, M., ZHAO, L., AND BADLER, N. I. 2000. The EMOTE model for effort and shape. In *Proc. SIGGRAPH 2000*. 173–182.

COHEN, M. AND MASSARO, D. 1993. Modeling coarticulation in synthetic visual speech. In *Models and Techniques in Computer Animation*, N. Magnenat-Thalmann and D. Thalmann, Eds. Springer, Tokyo, 139–156.

DE RUITER, J. 2000. The production of gesture and speech. In *Language and Gesture: Window into Thought and Action*, D. McNeill, Ed. Cambridge University Press, Cambridge, England; New York, NY, 284–311.

FALOUTSOS, P., VAN DE PANNE, M., AND TERZOPOULOS, D. 2001. The virtual stuntman: Dynamic characters with a repertoire of autonomous motor skills. *Computers & Graphics 25,* 6, 933–953.

FINKLER, W. AND NEUMANN, G. 1988. MORPHIX. A Fast Realization of a Classification-Based Approach to Morphology. In *4. Österreichische Artificial-Intelligence-Tagung. Wiener Workshop - Wissensbasierte Sprachverarbeitung. Proceedings.*, H. Trost, Ed. Springer, 11–19.

FREY, S. 1999. *Die Macht des Bildes: der Einfluß der nonverbalen Kommunikation auf Kultur und Politik*. Verlag Hans Huber, Bern.

HARTMANN, B., MANCINI, M., BUISINE, S., AND PELACHAUD, C. 2005. Design and evaluation of expressive gesture synthesis for embodied conversational agents. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems*. ACM Press.

HARTMANN, B., MANCINI, M., AND PELACHAUD, C. 2002. Formational parameters and adaptive prototype installation for MPEG-4 compliant gesture synthesis. In *Proc. Computer Animation 2002*. 111–119.

HARTMANN, B., MANCINI, M., AND PELACHAUD, C. 2006. Implementing expressive gesture synthesis for embodied conversational agents. In *Proc. Gesture Workshop 2005*. LNAI, vol. 3881. Springer, Berlin; Heidelberg, 45–55.

HODGINS, J. K., WOOTEN, W. L., BROGAN, D. C., AND O'BRIEN, J. F. 1995. Animating human athletics. In *Proc. SIGGRAPH 1995*. 71–78.

HOLLARS, M. G., ROSENTHAL, D. E., AND SHERMAN, M. A. 1994. *SD/FAST User's Manual*. Symbolic Dynamics Inc.

HUENERFAUTH, M., ZHOU, L., GU, E., AND ALLBECK, J. 2007. Design and evaluation of an american sign language generator. In *Proceedings of the Workshop on Embodied Language Processing*. Association for Computational Linguistics, Prague, Czech Republic, 51–58.

JURAFSKY, D. AND MARTIN, J. H. 2003. *Speech and Language Processing*. Prentice Hall.

KENDON, A. 1980. Gesticulation and speech: Two aspects of the process of utterance. In *The Relationship of Verbal and Nonverbal Communication*, M. Key, Ed. Mouton Publisher, The Hague, The Netherlands, 207–227.

KENDON, A. 2004. *Gesture – Visible Action as Utterance*. Cambridge University Press, Cambridge.

KIPP, M. 2001. Anvil – a Generic Annotation Tool for Multimodal Dialogue. In *Proceedings of the 7th European Conference on Speech Communication and Technology (Eurospeech)*. Aalborg, Denmark, 1367–1370.

KIPP, M. 2004. *Gesture Generation by Imitation: From Human Behavior to Computer Character Animation*. Dissertation.com, Boca Raton, Florida.

KIPP, M., NEFF, M., AND ALBRECHT, I. 2006. An Annotation Scheme for Conversational Gestures : How to economically capture timing and form. In *Proceedings of the Workshop on "Multimodal Corpora" at LREC 2006*. 24–27.

KIPP, M., NEFF, M., KIPP, K., AND ALBRECHT, I. 2007. Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In *Proceedings of Intelligent Virtual Agents (IVA07)*. LNAI, vol. 4722. Association for Computational Linguistics, 15–28.

KITA, S., VAN GIJN, I., AND VAN DER HULST, H. 1998. Movement phases in signs and co-speech gestures, and their transcription by human coders. In *Gesture and Sign Language in Human-Computer Interaction*, I. Wachsmuth and M. Fröhlich, Eds. Springer, Berlin, 23–35.

KOPP, S., SOWA, T., AND WACHSMUTH, I. 2004a. Imitation games with an artificial agent: From mimicking to understanding shape-related iconic gestures. In *Proc. Gesture Workshop 2003*. LNAI, vol. 2915. Springer, Berlin and Heidelberg, 436–447.

KOPP, S., TEPPER, P., AND CASSELL, J. 2004b. Towards integrated microplanning of language and iconic gesture for multimodal output. In *Proc. International Conference on Multimodal Interfaces 2004*. 97–104.

KOPP, S. AND WACHSMUTH, I. 2004c. Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds 15*, 39–52.

MANN, W. C. AND THOMPSON, S. A. 1988. Rhetorical Structure Theory: Toward a functional theory of text organization. *Text 8,* 3, 243–281.

MARTELL, C. H. 2004. Form: An extensible, kinematically based gesture annotation scheme. In *Natural, Intelligent and Effective Interaction in Multimodal Dialogue Systems*. Kluwer Academic Press.

MARTIN, J.-C., NIEWIADOMSKI, R., DEVILLERS, L., BUISINE, S., AND PELACHAUD, C. 2006. Multimodal Complex Emotions: Gesture Expressivity And Blended Facial Expressions. *Special issue of the Journal of Humanoid Robotics* 3 (September), 269–291.

MCNEILL, D. 1992. *Hand and Mind: What Gestures Reveal about Thought*. The University of Chicago Press, Chicago.

MCNEILL, D. 2005. *Gesture and Thought*. University of Chicago Press, Chicago.

MILLER, G. A., BECKWITH, R., FELBAUM, C., GROSS, D., AND MILLER, K. 1990. Introduction to WordNet: an on-line lexical database. *International Journal of Lexicography 3,* 4, 235–244.

NEFF, M. 2005. Aesthetic exploration and refinement: A computational framework for expressive character animation. Ph.D. Dissertation. Department of Computer Science, University of Toronto.

NEFF, M. AND FIUME, E. 2002. Modeling tension and relaxation for computer animation. In *Proc. ACM SIGGRAPH Symposium on Computer Animation 2002*. 81–88.

NEFF, M. AND FIUME, E. 2005. AER: Aesthetic Exploration and Refinement for expressive character animation. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2005*. 161–170.

NEFF, M. AND FIUME, E. 2006. Methods for exploring expressive stance. *Graphical Models 68,* 2, 133–157.

NEFF, M. AND SEIDEL, H.-P. 2006. Modeling relaxed hand shape for character animation. In *Articulated Motion and Deformable Objects (AMDO 2006)*. LNCS, vol. 4069. Springer, Berlin.

NOMA, T., ZHAO, L., AND BADLER, N. 2000. Design of a virtual human presenter. *IEEE Computer Graphics and Applications 20,* 4, 79–85.

NOOT, H. AND RUTTKAY, Z. 2004. Gesture in style. In *Proc. Gesture Workshop 2003*. LNAI, vol. 2915. Springer, Berlin; Heidelberg, 324–337.

POLLARD, N. S. AND ZORDAN, V. B. 2005. Physically based grasping control from example. In *Proc. ACM SIGGRAPH / Eurographics Symposium on Computer Animation 2005*. 311–318.

POPOVIC, Z. AND WITKIN, A. 1999. Physically based motion transformation. In *Proc. SIGGRAPH 1999*. 11–20.

PRESS, W. H., TUKOLSKY, S. A., VETTERLING, W. T., AND FLANNERY, B. P. 1992. *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. Cambridge University Press.

SAITZ, R. L. AND CERVENKA, E. J. 1972. *Handbook of Gestures: Colombia and the United States*, second ed. Mouton, The Hague.

SCHEFLEN, A. E. 1964. The Significance of Posture in Communication Systems. *Psychiatry 26*, 316–331.

SCHEGLOFF, E. A. 1984. On some gestures' relation to talk. In *Structures of Social Action*, J. M. Atkinson and J. Heritage, Eds. Cambridge University Press, Cambridge, 266–296.

SHAPIRO, A., FALOUTSOS, P., AND NG-THOW-HING, V. 2005. Dynamic animation and control environment. *Graphics Interface '05*, 61–70.

STEEDMAN, M. 2000. Information structure and the syntax-phonology interface. *Linguistic Inquiry 34*, 649–689.

STONE, M., DECARLO, D., OH, I., RODRIGUEZ, C., STERE, A., LEES, A., AND BREGLER, C. 2004. Speaking with hands: Creating animated conversational characters from recordings of human performance. In *Proc. SIGGRAPH 2004*. 506–513.

TAN, R. AND DAVIS, J. 2004. Differential video coding of face and gesture events in presentation videos. *Computer Vision and Image Understanding 96 (2)*, 200–215.

WEBB, R. 1997. *Linguistic Properties of Metaphoric Gestures*. UMI, New York.

WITKIN, A. AND KASS, M. 1988. Spacetime constraints. In *Proc. SIGGRAPH 1988*. 159–168.

WITKIN, A. AND POPOVIC, Z. 1995. Motion warping. *Proc. SIGGRAPH 1995*, 105–108.

ZORDAN, V. B. AND HODGINS, J. K. 2002. Motion capture-driven simulations that hit and react. In *Proc. ACM SIGGRAPH Symposium on Computer Animation 2002*. 89–96.