Augmenting Gesture Animation with Motion Capture Data to Provide Full-Body Engagement

Pengcheng Luo[1], Michael Kipp[2] and Michael Neff[1]

[1] UC Davis, USA, {pcluo|neff}@ucdavis.edu
[2] DFKI, Germany, michael.kipp@dfki.de

Abstract. Effective speakers engage their whole body when they gesture. It is difficult, however, to create such full body motion in animated agents while still supporting a large and flexible gesture set. This paper presents a hybrid system that combines motion capture data with a procedural animation system for arm gestures. Procedural approaches are well suited to supporting a large and easily modified set of gestures, but are less adept at producing subtle, full body movement. Our system aligns small motion capture samples of lower body movement, and procedurally generated spine rotation, with gesture strokes to create convincing full-body movement. A combined prediction model based on a Markov model and association rules is used to select these clips. Given basic information on the stroke, the system is fully automatic. A user study compares three cases: the model turned off, and two variants of our algorithm. Both versions of the model were shown to be preferable to no model and guidance is given on which variant is preferable.

Key words: Embodied Conversational Agents, Posture Synthesis, Motion Capture

1 Introduction

When creating virtual agents, the designer is caught between two main animation options, each with their inherent trade-offs. Procedural motion generation offers excellent control, allowing the agent to flexibly respond to a range of situations and generate a very large set of gestures. This flexibility, however, comes at the cost of extra work and/or realism as it is difficult to generate highly realistic motion using procedural methods. On the other hand, motion capture-based approaches provide an easier method to obtain realistic motion that engages the entire body, but control is generally limited.

In this paper, we present a hybrid system that uses procedural generation for arm gestures and motion capture data to add realistic body movement[3]. Our procedural methods for arm gesture are based on previously published techniques [1, 2]. The contribution of this work is a system for engaging the rest of the body,

[3] Animation samples can be found on http://www.cs.ucdavis.edu/~neff/pengcheng/

using motion capture data to control the lower body, augmented with procedural generation of body rotations. The approach leverages off the strengths of the two animation techniques: using procedural generation for arm gestures, where maximal control is necessary, while using motion capture to generate full body engagement and increase the realism of the final motion.

Inspired by Lamb's theory of body-gesture merger [3], discussed below, our algorithm aligns the character's body movement with the stroke phase of gestures to engage the total body in creating a gesture. The rules for this alignment are based on a statistical model built from a sample speaker. It uses both a Markov model and association rules to predict a desired weight shift and body orientation for each stroke. The system then searches for a short piece of motion capture data to satisfy these goals and uses it to control the lower body movement during the stroke. Rotation of the spine is also added procedurally. We evaluate our model in a user study that compares three cases: our body-engagement model turned off and two variants of our model, one that uses motion capture to in-fill body motion between strokes and one that uses interpolation and hold phases for this in-fill. The study showed with high significance that both models were preferable to no model. It also showed that the motion capture gap filling model was preferable to the gap filling based on interpolation. Sample frames from animation produced by our system are shown in Figure 1.

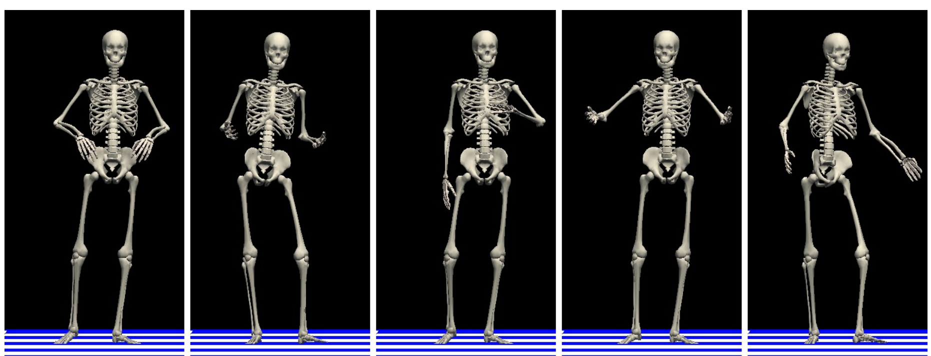

Fig. 1. Sample frames showing our algorithm used to add postural movement to a gesture sequence.

This paper makes two main contributions. First, it presents an effective algorithm for automatically augmenting animations of arm gestures with appropriate postural movement. The resulting system allows a very wide range of gestures to be generated with a higher degree of naturalness than was previously possible. The algorithm is validated with a user study. The second contribution is a study that indicates a strong preference for the continuous use of motion capture data rather than a mix of motion capture data with interpolation and short holds.

This suggests that continuous movement may be an important factor in the generating positive perceptions of virtual agents.

2 Background

2.1 Gesture and Posture Study

In his pioneering work, Lamb [3] argued that there were two broad classes of movers, those that made arm gestures largely independently of postural changes and those that effectively merged posture changes with gesture. He argued that this *posture-gesture merger* in the latter group led to much more effective movement. This idea provided the key motivation for aligning posture changes with gesture strokes in our model.

Cassell et al. [4] conducted an analysis of posture shifts in monologue and dialogues. They predicted posture shift as a function of discourse state in monologues and discourse state and conversation state in dialogues. We instead focus on more ongoing motion aligned with gesture strokes.

Gesture has been previously described in terms of phases, phrases and units [5–8], where a single gesture consists of a series of consecutive phases:

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

We use this same structure in our work and in particular, use the stroke placement to align our posture movement. Previous work has used this structure in recreating gesture animation [2, 1].

For statistical modeling, Lee and Marsella [9] used a Hidden Markov model to generate head movements for virtual agents. In a very different domain, Khalil et al. [10] performed statistical modeling that combined association rules and a Markov model to predict web users' behavior and their next movement.

2.2 Animation Methods

There are a diverse set of papers on using motion capture in animation. A few of the most relevant works include Arikan et al.s'[11] synthesis of motions by controlling the qualitative annotations of motion clips. Pullen and Bregler [12] developed a conceptually similar approach for creating motions by considering the motion capture data as a texture to be applied to more simple key framed animation, but did not apply their work to gesturing characters, made no use of gesture structure and used a very different formulation based on a frequency decomposition. Arikan et al. [13] and Kovar et al. [14] proposed similar methods using motion graphs to connect motion clips together. Wang and Bodenheimer [15] studied the best parameters to use when doing linear blends to connect motion clips together.

Stone et al. [16] present a system for using motion capture to animate a conversational agent. They use motion capture to control the entire body, whereas our approach combines motion capture with a more flexible, procedural approach for gesture generation.

The procedural animation system for gestures follows the work of Neff et al. [1] which combines a statistical method for predicting gesture selection and placement with a procedural animation system to create an animated character that could gesture in synchrony with speech. In related work, the SmartBody engine [17] offers another way to combine different animation modalities.

3 Acquiring and Analyzing Input Motion

The algorithm augments procedural arm gestures with lower body motion and rotation. The rules for doing this are determined by analyzing sample data of a subject whose body engagement is considered desirable. This input data is used both in determining these rules and as the source motion data that is used in the reconstruction.

For our input data, we had a subject recite several versions of Shakespeare's famous Marc Antony speech "Friends, Romans, Countrymen..." while being both filmed and motion captured with an optical motion capture system [18]. This provided data of a long monologue, which we analyzed both in terms of speech and motion data. We manually annotated the data using the software package ANVIL [19]. This annotation included gesture *phrase*, gesture *phase*, the number of stroke repetitions and the hand used in each phrase (right hand, left hand or both hands). A trained coder can annotate 1 minute of video in ca. 25 mins.

The motion analysis began by reconstructing an animation skeleton from the motion capture data to obtain joint angle data. This data was then processed to extract a number of key parameters that were hypothesized to be important in lower body movement: center of mass (COM) and foot locations, swivel angles of the legs, pelvic rotation, and knee angles. A correlation analysis was performed across this data. It revealed that, taken together, the COM position and rotation of the pelvis around the vertical axis ($pelvis_Y$) were well correlated with the other data, having a correlation coefficient higher than 0.6. Some parameters correlated well with COM and others with pelvic rotation. Figure 2 shows an example of how the leg swivel data is correlated with the pelvis rotation. Based on this analysis, we determined that these two parameters effectively characterized the major lower body motion. We therefore focused on accurately reconstructing them in order to add lower body motion to new clips.

4 Statistical model for posture prediction

For the domain of talking characters, lower body movement can be divided into two categories: movement co-occurring with gesture and movement not co-occuring with gesture, also known as idle movement. We segment the lower body motion based on this definition and use the posture information at the end of each gesture phrase for analysis and modeling.

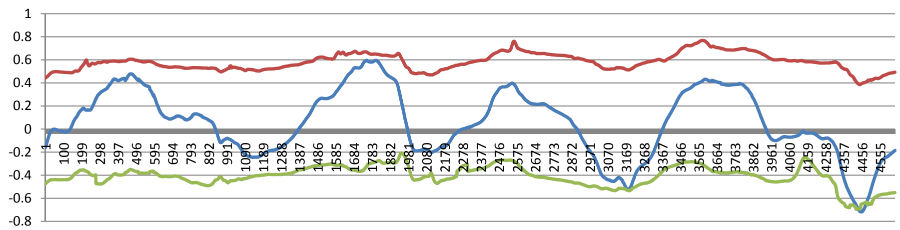

Fig. 2. In a plot of leg swivel data and pelvis data for a motion clip, a clear correlation can be seen between the data. Top: $Pelvis_Y$; middle: $LeftFootSwivel$; bottom: $RightFootSwivel$.

4.1 Markov Model

Posture configurations can be roughly defined into three categories for each of our posture parameters. For the center of mass, it contains the categories: weight shifted to the left foot, weight shifted to the right foot, and balance in the center of the two feet. The pelvis rotation around the vertical axis has three directions: facing left, facing right and facing front. Intuitively, having a variety of posture positions at the end of phrases makes motions more realistic. Imagine if a character consistently returns back to the exact same position at the end of a gesture, the motion will not be realistic. While using only three categories per parameter, our system will support a range of end poses within each category to avoid repetitive motion.

A Markov model is commonly used in modeling time series sequences. For example, Kipp et al. [2, 1] used Markov models for modeling handedness and gesture sequences. Inspired by their work, we use a Markov approach for modeling the continuous posture sequence information. Let X be the end posture at each phrase, and n stands for the index of the phrase in the whole sequence. For a Markov chain:

$$P(X_{n+1}|X_n, X_{n-1}, \cdots) = P(X_{n+1}|X_n) \tag{2}$$

Or in words, the future state depends only on the present state and the past can be ignored. While this model captures some of the modeled individual's tendency to make weight shifts or rotations, it is not complete. For instance, it does not take into consideration the choice of synchronizing particular postures with given gestures. We turn our attention to that next.

4.2 Association Rules

We postulated that a person's posture changes (rotation and weight shifts) would be related to the form of the gesture they were performing. Different gestures would potentially lead to different posture changes. For example, a person may tend to shift his weight right when using his right hand. Modeling these properties has two benefits. It provides a better choice of posture change for a given gesture. It also adds more variation into the transition probability from one posture to the next. This will provide greater variation over a sequence.

Association rules are one of the most important concepts in data mining and commonly used in finding association patterns in a series of data sequences. Given a set of transactions, association rules will find useful hidden rules, given some threshold, to help predict values in the future. Generally speaking, there are two important metrics: one is the support and the other is the confidence. Suppose there are two itemsets X and Y where an itemset is just a set of observed attributes (e.g. a left body rotation with a left-handed cup gesture and no weight shift). The support count of X is the fraction of transactions in the database that contain X. Confidence measures how often items in Y appear in transactions that contain X and can be interpreted as an estimate of $P(Y|X)$.

We use association rules to model the relationship between gesture types and posture positions at the end of gestures. Each gesture phrase is a transaction in our model which contains the information of handedness (H), gesture lexeme (L) (e.g. a "cup" gesture or "frame" gesture) and an end of phrase posture description including the center of mass position (C) and the direction of body rotation (B).

To calculate the confidence of the center of mass position, we have the following formulas.

$Support(H) = count(H)/totalcount$
$Support(L) = count(L)/totalcount$
$Support(C) = count(C)/totalcount$
$Confidence(C|H) = count(C \cap H)/count(H)$
$Confidence(C|L) = count(C \cap L)/count(L)$
$Confidence(C|H, L) = count(C \cap H \cap L)/count(H \cap L)$

Where totalcount is the number of transactions appearing in the database.

Since there are many gesture lexemes and three handedness type, we will find a variety of association rules, but not all of them will have enough data support to be meaningful. Thus we first used a threshold filter on the itemsets to remove those that have too small a support number and are not considered meaningful. We then set up another threshold to filter the association rules whose confidence is small. The same strategy is used for prediction rules for body rotation. In the end, we are left with a list of useful association rules for each category of COM position and body rotation direction with satisfactory support and confidence values.

4.3 Combining Prediction Rules

The final prediction of the body movement is made by combining the output of the Markov model and the association rules. The input to the system is the initial center of mass position (IC), the gesture lexeme (L) and Handedness (H). We will get the probability of transition from IC to three categories of center of mass using the probability PM calculated from Markov model. In addition, we take the highest confidence value from our association rules and denote that as PC, which also describe the probability of transition from IC to the three categories of center of mass at the end of the phrase. The final probability distribution is defined as:

$$P = \alpha PM + \beta PC \tag{3}$$

where β was experimentally set to between 0.6 and 0.7 and $\alpha = 1 - \beta$. The same approach is used in predicting final body rotation direction. Since center of mass has three possibilities, and body rotation has three possibility, thus in total there are 9 possibilities in describing the final posture. We have a probability distribution that gives the odds of each of these occurring and we randomly sample from this distribution to choose the actual posture change. By changing our random seed, we can produce different motion sequences from a given distribution.

5 Animation Methods

5.1 Motion Capture

Motion capture has recently become a widely used alternative to keyframe or procedural techniques in the animation community. Compared with these traditional methods, motion capture will provide more detailed motions which were difficult to model previously. However few papers have addressed how to use this method to model lower body motion. Egges et al. [20] built up a general framework to insert idle movement into general animation. We use similar approaches in modeling idle movement but do so in a different reconstruction framework and within a larger system that focuses on aligning body motion with gesture.

5.2 Selection and Reconstruction of Co-occurring Motion Clips

The reconstruction method works in two stages. First, it finds appropriate motion capture clips to align with each stroke, as described below. Second, it fill in the gaps between these clips, as described in the following subsection.

The motion capture data used to generate posture movement is first preprocessed. The motions are divided into segments that co-occur with each gesture phrase in the sample and assigned to 9 groups based on their center of mass and pelvis rotation information. The motion samples used in the reconstruction are chosen one-by-one from this set according to the motion prediction rules to align lower body movement with co-occurring gestures. The motions are reconstructed as follows:

First, lower body motion clips are aligned with the end of stroke. Before working with our data, a butterworth filter is applied to the motion to smooth out small jerks. For our sample data, if there is an extreme value (minimum or maximum) within the phrase we then calculated the time difference between extreme values, where the velocity equals 0, and the end of the stroke. We found that the extreme for the pelvis rotation is usually between 0 and 10 percent of the average phrase duration before the end of the stroke. The end of the horizontal COM shift is usually between 0 and 20 percent of the average phrase time after the end of the stroke. This means that the pelvis usually stops a little before the end of stroke and the center of mass usually stops a little bit after the end of stroke which supports our assumption that the posture change usually stops around the end of stroke if it is going to pause within the phrase. While interesting, these conclusions are drawn on limited data for one speaker, so general conclusions about human movement should not be drawn.

Since there are many candidate motions for each clip selection, we select motions whose starting point has the smallest velocity difference from the average velocity over the idle duration as illustrated in Figure 3. The difference is calculated by

$$Diff = V_S - V_A \tag{4}$$

where $Diff$ is the difference, V_S is the velocity at the starting point of the phrase for the next clip and V_A is the average velocity of the idle movement. The velocities are the summed difference velocities of the COM and pelvis rotation. Ensuring a small velocity difference helps ensure realistic motion.

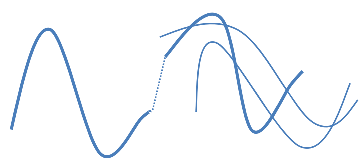

Fig. 3. Clips are chosen to minimize the velocity difference in order to help produce smooth motion.

In some cases, two consecutive selected motions will have a large gap between their end points that must be spanned in a short time. This can lead to unrealistically rapid movement. Instead, we find a path from the end of the previous motion to a point further ahead in time on the next motion that is closer to the previous motion's end state. This case is shown in Figure 4. The adjustment produces smoother overall motion.

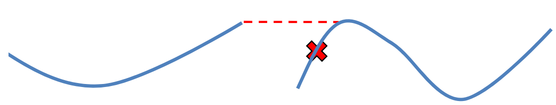

Fig. 4. For rapid transitions, the next clip, shown in blue is truncated to allow a more direct transition from the previous clip.

Sometimes, the postures predicted for the end of consecutive phrases will fall in the same category. In order to create more continuous motion in these cases, the algorithm finds longer, continuous motion clips from the mocap data to fit this sequence of motions.

5.3 Idle Motion Reconstruction

Fig. 5. Once motion clips, the thick blue curves, are aligned with the gestures, the remaining gaps must be filled using either interpolatuion or additional clips, as represented by the thin red curve.

At this point, the system has a partial specification of lower body movement, with short clips aligned with each gesture stroke and gaps in between. The task now is to fill these gaps, the so-called *idle movement*, as shown in Figure 5. We designed two strategies for this. The first is to fill the gaps by selecting appropriate mocap clips and blending them into the overall motion stream. The second is to use a combination interpolation and short holds where the interpolation would be too slow. The effectiveness of the two strategies is compared in the user study presented in Section 6. The strategies are described below.

Motion Capture Gap Filling When filling gaps with motion capture clips, we search the mocap database to find the correct duration of motion to fit the idle gap. Motion clips are selected based on their distance from the motions that have already been specified at the two ends of the gap. A sliding window is used to define how many frames are compared. Let the length of the window be L, and the motion clips already aligned with the gestures are represented by $M_{1,}M_{2,}\cdots M_m$, the source original motion clips are represented by $M^{'}_{1,}M^{'}_{2,}\cdots M^{'}_n$. The distance

$$D(k,j,f,L) = \sum_{i=0}^{L} \alpha(M_{k-i} - M^{'}_{j-i}) + \beta(M_{k+i+1} - M^{'}_{j+f+i+1}) \qquad (5)$$

where f is the number of frames in the gap, i is the index over the frames being compared, k is the start frame in the selected motion and j is the start frame in the source motion. α and β are adjusted to give greater weight to frames close to the gap and less weight further away. The sliding window ranges from 10 to 30 frames. This comparison combines the velocity comparison and distance comparison, which makes the selected motion the most similar to the motion on either end.

When motion clips are selected, we have to register the selected motion to the gaps, aligning them with the motion at either end. This is done by adding a linear offset to the fill clip which generates a good result. This can be represented formally as $F(x) = G(x) + O(x)$, where $O(x)$ is the linear offset, $G(x)$ is the selected motion clips, $F(x)$ is the final fitted motion for a frame index x. $G(x)$ and $O(x)$ have the same starting and ending index. Since $G(x)$ and $O(x)$ are both continuous functions, thus the final motion $F(x)$ is also continuous. However, sometimes the added offset will make the motion unrealistic by producing overly large movements. The sequence $F(x)$ can be multiplied by a scale factor to reduce the final motion to a more realistic range.

Interpolation and Hold Gap Filling Another strategy explored is using interpolation to fill the gaps. The motivation for this strategy is to use the most complex movement for the stroke and relatively simple motion in between, the hope being that this would add greater emphasis to the stroke motion, where the communicative meaning is concentrated. The interpolation fill is combined with inserting short hold phases. These serve two purposes. First, they reflect the periods of stillness observed in our data. Second, they provide a way of ensuring the velocity of the movement is not unrealistically slow. If the gap requires a long duration interpolation with a relatively small change in posture, this would cause unrealistically slow movement. A hold phase is used to occupy some of the gap time, so the actual transition proceeds at a reasonable velocity.

This fill scheme is implemented as follows. To avoid a sudden stop in the motion, we extend the end point of the phrase clip into the gap by an amount $t = s_1 * GapDuration$ where s_1 is a scale factor set to 0.25. The new end position is calculated by multiplying the scaled velocity at the end of the clip ($V = 0.08*v$) by this time where v is the velocity at the end of last phrase. This allows the motion to fade out. A hold is then started at $HoldStart = p + V * t$ where p is the position at the end of last Phrase and the hold duration is $t' = s_2 * GapDuration$ where s_2 is a scale factor set to 0.25. The interpolation between the motion clips and hold is done using an ease-in, ease-out curve to connect the ends of the hold to the previous and next phrases.

5.4 Procedural Animation

Procedural animation is used to add body rotation to the motion. This is done by specifying an axial rotation along the spine. The desired body rotation direction is determined based on the same prediction used for the pelvis rotation. To determine the starting time for the rotation, we calculate the time difference between the phrase start time and the start time of body rotation for all samples in our source data. This offset can be modeled using a Gaussian distribution that has average AVE and standard deviation SD. During reconstruction, we sample from this distribution to determine the time difference between the starting point of phrase and the start of the rotation. The end time and the end value for body rotation is set up using default values which is similar to the approach in [1].

5.5 Motion Reconstruction

The approach described here is general enough to be adapted to various animation systems. Our specific animation system reconstructs the motion as follows, given as input a script that specifies the desired gestures and associated timings. First, the system solves for the arm posture associated with the start and end of each stroke. Interpolation functions are used to move between poses. Additional data such as hand-shape is also specified at this time, following the algorithm described by [1]. A similar approach is used for the procedural body rotation. Start and end poses for the effected spine degrees of freedom are specified and interpolation functions are associated with them to affect the transition. The algorithms described above generate a continuous sequence of motion capture data for controlling the lower body. Instead of representing this as joint angle data, our system stores this as a set of parameters that can be used to reconstruct the pose at each time step. Parameters include the COM position, foot positions, knee angles and pelvis angles. At each time step, a dedicated lower body solver uses this data to pose the lower body as the upper body is controlled by interpolating between the keyed values. Footskate is corrected by specifying periods during which each foot cannot move and then simply keeping the foot data constant during these times.

6 Evaluation

The algorithm provides visually pleasing motion that greatly adds to the overall liveliness of the animation. The accompanying video provides examples of the algorithm, including the motion capture fill and interpolation fill variants.

We conducted a user study to test the effects of our system. First, we wanted to make sure that adding body movements is an improvement at all. Second, we wanted to empirically find out which of the two variants of our model produces more natural motion. For our study we recruited 21 subjects (14 male, 7 female) from the US (11) and Germany (10), aged 23–46.

Material We used a single clip from previous work of length 33 sec. to test different conditions [2]. Our conditions were: (N) no body motion, (F) body motion generated with the motion capture fill-in variant of our system, and (I) body motion with the interpolation variant. To gather sufficient power for the study we produced 4 different variations of each of conditions F and I using different random seeds in our probabilistic algorithm. We then cut the clips into 3 parts each. This resulted in 3 clips for N and 4x3=12 clips for F and I each. We intentionally left the audio track (speech) in the material because we considered the multimodal synchronization between speech, gesture and pose to be an important aspect for judging the naturalness of the motion.

Method Each subject participated in two studies (A and B). In study A we presented *single clips* in random order. We used 12 clips for F, I and N each[4] So the user was exposed to 3x12 = 36 clips in total, clips could not be replayed. The user rated the "naturalness" of the motion on a 5-point scale where every option was numbered (-2 to +2) and the extremes labeled with *not at all* and *very much*. In study B, we presented all 12 clips of conditions F and I *side by side*, in random left-right order. The user was asked to decide which variant s/he found more "natural" in terms of movement. Below the two videos, which could be replayed multiple times, we displayed a fully labeled 5-point scale: -2 (left one), -1 (rather left one), 0 (both equal), +1 (rather right one), +2 (right one).

Results In study A, the subjects rated the three conditions on average: -0.63 (sd=.68) for no motion (N), +1.1 (sd=.38) for motion capture fill (M), and +0.65 (sd=.41)for interpolated fill (I). We found that conditions F and I were rated more natural than N (no motion) with high significance. For this, we used t-tests for N vs. F (t(40)=-10.19; $p < .001$) and N vs. I (t(40)=-7.36; $p < .001$). Moreover, condition F was clearly rated more natural than I (t(40)=3.75; $p < .001$). The latter was confirmed by study B where we mapped ratings such that F corresponded to -2 and I to +2. The mean of -0.73 indicated that users preferred F which was validated to be a significant deviation from zero in a one-sample t-test (t(20)=-5.78; $p < .001$).

Discussion The highly significant preference of our proposed models to no motion (N) first ensures that our methods improve an animation instead of adding irritation. This finding underlines the importance of adding lower body motion at all in order to make an agent believable. The highly significant preference of I over F demonstrates how systematic studies can guide the algorithm design process and suggests the use of motion capture fill is a preferable option.

7 Conclusion

This paper presents an effective algorithm for adding full body postural movement to animation sequences of arm gestures. The system uses motion capture data and procedural animation to add lower body movement and spine rotation respectively. A combination of a Markov model and association rules are used to predict appropriate postural movement for specified gestures. The resulting motion has a much more lively, realistic feel while still maintaining the flexibility of a procedural gesture system capable of creating 38 different types of gestures in hundreds of variations.

A user study confirms the effectiveness of the algorithm. Two variants of the algorithm, using gap filling based on motion capture and gap filling based on interpolation, were both considered significantly preferable to not having a postural model. Interestingly, the motion capture gap filling was also considered

[4] To obtain 12 clips for condition N, we had to repeat each of the 3 clips 4 times.

significantly preferable to the interpolation based method. This suggests that continuous movement may be an important factor in subjects' judgement of animated motion.

Acknowledgments. We would like to thank Alexis Heloir and Tyler Martin for motion capture support. Financial support for part of this work was provided by the Department of Computer Science, UC Davis. Part of this research was conducted in the framework of the Excellence Cluster Multimodal Computing and Interaction (MMCI), sponsored by the German Research Foundation (DFG).

References

1. Neff, M., Kipp, M., Albrecht, I., Seidel, H.P.: Gesture modeling and animation based on a probabilistic re-creation of speaker style. ACM Transactions on Graphics **27**(1) (March 2008) 5:1–5:24
2. Kipp, M., Neff, M., Kipp, K., Albrecht, I.: Towards natural gesture synthesis: Evaluating gesture units in a data-driven approach to gesture synthesis. In: Proceedings of Intelligent Virtual Agents (IVA07). Volume 4722 of LNAI., Association for Computational Linguistics (2007) 15–28
3. Lamb, W.: Posture and gesture: an introduction to the study of physical behavior. Duckworth, London (1965)
4. Cassell, J., Nakano, Y., Bickmore, T., Sidner, C., Rich, C.: Annotating and generating posture from discourse structure in embodied conversational agents. In: Workshop on Representing, Annotating, and Evaluating Non-Verbal and Verbal Communicative Acts to Achieve Contextual Embodied Agents, Autonomous Agents 2001 Conference. (2001)
5. McNeill, D.: Hand and Mind: What Gestures Reveal about Thought. University of Chicago Press, Chicago (1992)
6. Kita, S., van Gijn, I., van der Hulst, H.: Movement phases in signs and co-speech gestures, and their transcription by human coders. In Wachsmuth, I., Fröhlich, M., eds.: Gesture and Sign Language in Human-Computer Interaction, Berlin, Springer (1998) 23–35
7. Kendon, A.: Gesture – Visible Action as Utterance. Cambridge University Press, Cambridge (2004)
8. McNeill, D.: Gesture and Thought. University of Chicago Press, Chicago (2005)
9. Lee, J., Marsella, S.: Learning a Model of Speaker Head Nods using Gesture Corpora. In: 7th International Conference on Autonomous Agents and Multi-Agent Systems, Budapest, Hungary. (2009)
10. Faten Khalil, J.L., Wang, H.: A framework of combining markov model with association rules for predicting web page accesses. In: Proc. Fifth Australasian Data Mining Conference, AusDM2006. (2006) 177–184
11. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. ACM Transactions on Graphics **22**(3) (2003) 402–408
12. Pullen, K., Bregler, C.: Motion capture assisted animation: Texturing and synthesis. ACM Transactions on Graphics **21**(3) (2002) 501–508
13. Arikan, O., Forsyth, D.A.: Interactive motion generation from examples. ACM Transactions on Graphics **21**(3) (2002) 483–490

14. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Transactions on Graphics **21**(3) (2002) 473–482
15. Wang, J., Bodenheimer, B.: Synthesis and evaluation of linear motion transitions. ACM Transactions on Graphics **27**(1) (March 2008) 1:1–1:15
16. Stone, M., DeCarlo, D., Oh, I., Rodriguez, C., Stere, A., Lees, A., Bregler, C.: Speaking with hands: creating animated conversational characters from recordings of human performance. ACM Transactions on Graphics **23**(3) (2004) 506–513
17. Thiebaux, M., Marshall, A., Marsella, S., Kallman, M.: Smartbody: Behavior realization for embodied conversational agents. In: Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008). (2008)
18. : (2008) Vicon, Inc. http://www.vicon.com.
19. Kipp, M.: Anvil: The video annotation research tool (2008) http://www.anvil-software.de/.
20. Egges, A., Molet, T., Magnenat-Thalmann, N.: Personalised real-time idle motion synthesis. In: 12th Pacific Conference on Computer Graphics and Applications. (October 2004) 121–130