BodySpeech: A configurable facial and gesture animation system for speaking avatars

A. Fernández-Baena, M. Antonijoan, R. Montaño, A. Fusté, and J. Amores
Grup de Tecnologies Mèdia (GTM), La Salle - Universitat Ramon Llull, Barcelona, Catalonia, Spain

Abstract— *Speaking avatars are present in many Human Computer Interaction (HCI) applications. Their importance lies in communicative goals which entail interaction within other avatars in virtual worlds or in marketing where they have become useful in customer push strategies. Generating automatic and plausible animations from speech cues have become a challenge. We present BodySpeech, an automatic system to generate gesture and facial animations driven by speech. Body gestures are aligned with pitch accents and selected based on the strength relation between speech and body gestures. Concurrently, facial animation is generated for lip sync, adding emphatic hints according to intonation strength. Furthermore, we have implemented a tool for animators. This tool enables us to modify the detection of pitch accents and the intonation strength influence on output animations, allowing animators to define the activation of gestural performances.*

Keywords: human computer interaction; speaking avatars; gesture animation; facial animation

1. Introduction

Face-to-face communication has the goal of transmitting a message from one person to another. Besides the semantics, the way a message is transmitted can change how the receiver perceives it. Body language and facial animations accompany the acoustic signal of speech, and moreover, they enrich communication and make it believable [1]. So, in order to make human computer interfaces believable, we must take into account the characteristics of the visual speech. Given the difficulty of creating realistic speech animations automatically, many companies use hand-crafted animations. Generating specific animations for any speech utterance results in increased production time and budget. On the other hand, automatic synthesis of gestures according to speech have been broadly studied in the character animation research community [2][3], providing a solution for the mentioned issues.

In this paper we present BodySpeech: an automatic method to generate appropriate body gestures and facial expressions according to an arbitrary speech. The system is able to select body gestures based on speech intonation and to concatenate them generating a smooth motion stream. We use mocap data to create a motion graph [4] which is named gesture motion graph (GMG). The animation system generates a continuous stream of gestures by concatenating units included in the GMG. The gesture selection process is driven by prosodic features of pitch accents (changes in speech intonation) in speech. Pitch accents and their corresponding features (time and strength) are automatically detected based on [5]. For every pitch accent, the system selects a gesture phrase with an equivalent strength (see Figure 1). Moreover, gestures and pitch accents are aligned in time. At same time, facial animation is generated by lip sync. We use the blendshapes approach to create visemes (facial shapes) that are assigned to phonemes. Additionally, we modify output visemes based on pitch accent strength for each pitch accent.

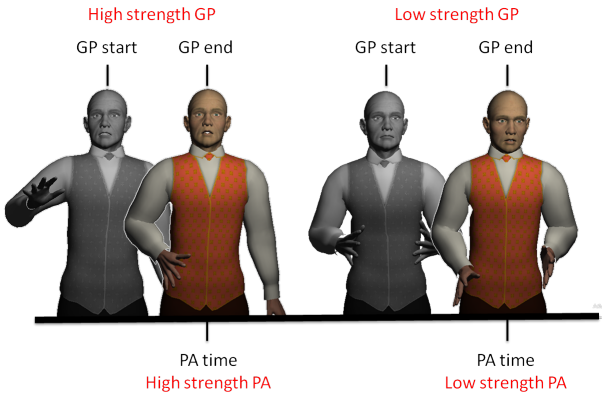

Fig. 1: BodySpeech. Gestural phrases (GP) are matched with pitch accents (PA) times and strength.

Moreover, we have implemented an application for animators that makes the generation of gesture and facial animations for speaking avatars easier. Using this application, animators can modify the speaker style by changing emphasis parameters. Emphasis of output animations depends on the frequency of performed gestures and the kinematic features of gestural movements. So, we facilitate the parameterized detection of pitch accents and determine how their strength affects output body and facial animations.

We summarize the related work in Section 2. Then, we describe the BodySpeech system in Section 3. Finally, we present an application for animators (Section 4) and conclude our work in Section 5.

This work expands on our prior work [6], primarily in

the automatization of gesture synthesis. In this paper we consider the automatic detection of pitch accents avoiding a manual annotation phase. This new way of detecting pitch accents is accompanied by a new pitch accent strength computation (more details in Section 3.3.1). Thanks to this, we have defined a new gesture-speech strength relation used in gesture selection (3.3.2.2). In terms of animation, we have enriched the GMG by reusing the input data to create more gestures (3.2.1). In addition, in order to avoid stroke modification (the most meaningful part of gestures) we have improved gesture temporal alignment with speech (3.3.2.1). Moreover, to improve output motion quality we use optimal blend length [7] to create blended transitions between gestures. Furthermore, as we have mentioned, we have added facial animation and implemented an authoring animation tool.

2. Related work

The generation of appropriate body language to a specific speech stream is a complex task. It is known that speech and gestures are related [8][9][10]. However, it is difficult to extract a set of rules capable of covering the broad variety of gestures taxonomy [1] (iconic, metaphoric, deictic and beats) and then to use that information to drive a gesture synthesis system. Another challenge arises from the attempt to automatize the gesture selection and animation synthesis processes, avoiding the time-consuming step of manual annotation.

One early attempt to generate body language automatically was presented by Casell et al. in BEAT [11]. They presented a system that analyzes an input text (natural language structure and content), and defines a set of gesture generation algorithms that suggest gestures depending on the result of the text analysis. The algorithms rely on a manually created Knowledge Base, which defines the gestures that are appropriate to certain actions or objects. Stone et al. [12] presented another automatic gesture synthesizer. However, in this case it uses a unit selection approach, and units are pieces of motion captured from real performances. This permits the generation of animation that naturally contains the subtleties of real human motion, which are hard to reproduce otherwise. Stone's synthesizer is limited to generating utterances present in a pre-defined grammar. Although this grammar can be extended as much as desired, the creation of this grammar requires some manual annotation. Neff et al. [13] proposed a novel system, that from an input text is capable of generating animations that recreate the style of a certain speaker. The process begins with a gesture selection step, which is driven by a statistical model created from performances of the speaker. In the next step the animation engine uses parameters that define the shape of gestures produced by the speaker and a set of predefined rules to produce the animation. The system is fully automatic but requires some annotation in the input text.

Other systems, do not rely in input text to generate animation but in prosodic parameters of speech directly. This allows to go a step further in adaptability because these parameters can be extracted either from the output of a text-to-speech synthesizer, as well from the audio of real speech. A limitation of these systems is that it is not possible to extract language structure or semantic content from prosody, and therefore they cannot be correlated with content of gestures. Moreover, prosodic-based gesture synthesizers usually only generate beat type gestures. Beats are a type of gesture that do not carry meaning, and their function is to emphasize words in a utterance [1]. It is known that prosody correlates well with emphasis [14], which suggests that beats are good candidates to be synthesized based on prosody. Levine et al. proposed two algorithms [15] [2] that automatically generate beat gestures based on input audio. Their systems use statistical models that shape the correlations between prosody and kinematic parameters of gestures. These models are used to select gestural units stored in a mocap database. Gestural units are composed of a single gestural phase. In a further work, Chiu et al. [3] presented a similar system but in this case units are composed by single animation frames. This permits the generation of a greater variety of gestures at cost of animation realism. Our approach is similar to Chiu's and Levine's in the fact that it uses prosody to select motion units from a database. However, the unit selection process is not governed by a statistical model but by a set of rules. This allows greater parametrization of the process, which in turn provides greater control of the output.

3. BodySpeech

3.1 Overview

The animation system is divided into two stages: an offline preprocessing step and a runtime unsupervised step. Figure 2 shows an outline of the whole system. In the first stage, gesture mocap data is arranged in a motion graph structure as described in Section 3.2.1. On the other hand, we associate visemes (mouth shapes) with phonemes. Vowel phonemes have more than one associated viseme in order to capture emphasis in facial animation. Visemes parameterization is further explained in Section 3.2.2.

The second stage is where the output animation is generated. Speech is used to drive both gesture synthesis and facial animation. Input speech is analyzed in order to detect pitch accents (time occurrence and strength indicator) and the phoneme transcription of the message (Section 3.3.1). Pitch accents drive gesture synthesis by selecting the most appropriate gesture unit for each one depending on strength levels (see more details in Section 3.3.2). We use gestural phrases as gesture units. A gestural phrase [16] consists of the following phases: stroke (obligatory phase where it is contained the 'expression of the gesture'), preparation (movement that leads to the beginning of the stroke) and

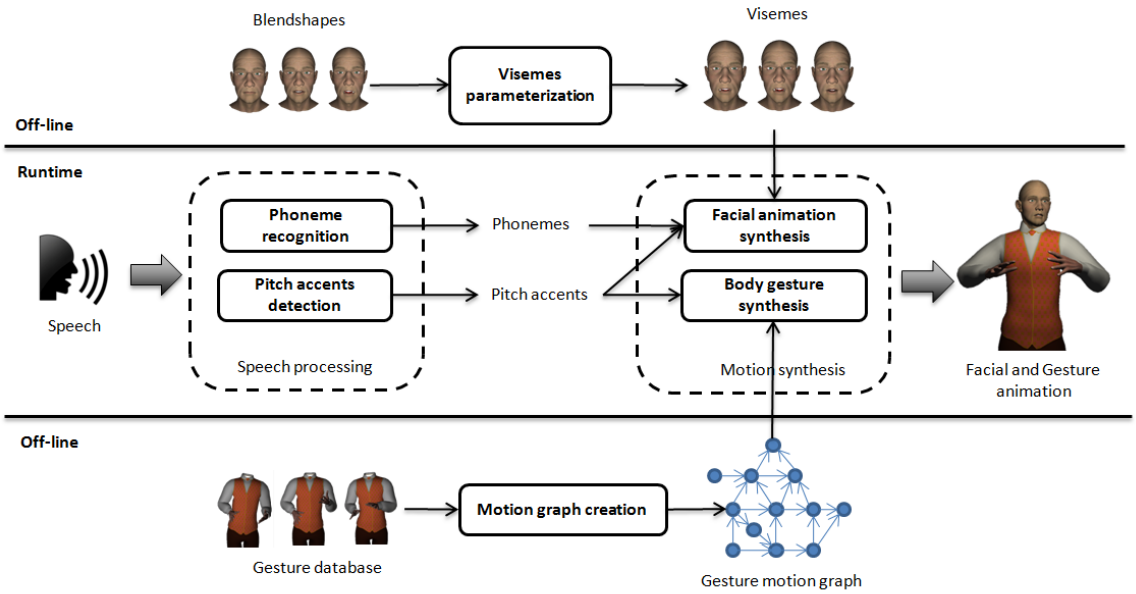

Fig. 2: System overview. The off-line stage is used to generate a motion graph (at the bottom) and a set of visemes (on top). Then, body and facial animation are obtained from a speech signal in the runtime stage (on the middle).

retraction (body parts are moved to the rest position). Moreover, gestural phrases may include hold phases which are temporary cessations of movement. At the same time, phonemes intervals are matched with visemes to generate facial animation. Additionally, visemes are modified based on pitch accent strength indicators (see Section 3.3.3).

3.2 Off-line stage

3.2.1 Motion graph creation

A labeled gesture motion database is used to construct a motion graph [4]. This database consists of 6 clips that last slightly more than one minute each, in which an amateur actor with mocap recording experience was asked to perform an improvised monologue with a concrete speaking style and performing only beat gestures. We choose neutral and aggressive style in order to obtain a broad variety of gestures with different strengths. Gestural phrases and their corresponding gesture phases are annotated in this database. Also, stroke apexes (the maximum extension point) are annotated. So, we use gestural phrases (GP) motion clips to populate a gesture motion graph (GMG). Also, stroke phases are extracted and added as new gestural phrases. In this way, we maximize the number of gestural phrases allowing more variety in gesture synthesis.

A GMG is defined by N for the set of nodes (GP's), E for the set of edges (transitions) and $W(E)$ for the set of edge weights (transition parameters). First, we connect all the consecutive GP's from the original database with a directional edge. Then, we create new edges between non-consecutive GP's. Transitioning between non consecutive

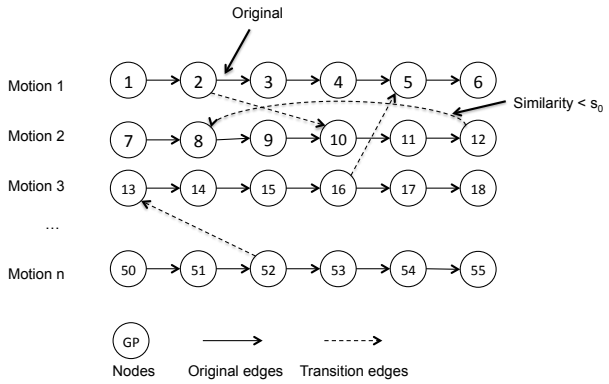

Fig. 3: Edge generation in gesture motion graph. We create an edge connecting two gestural phrases if they are consecutive in the original recordings (original edges) or their posture similarity is below s_0 (transition edges).

gesture phrases can produce jerky motions if GPs extreme postures are not similar enough. Therefore, we compute posture similarity between initial and ending frames from all motion clips in the graph using joint angles distance metric [4]. As a consequence, we create edges when the similarity value is lower than a threshold s_0 (see Figure 3). In order to search the appropriate gestures in the motion graph and to generate smooth transitions between GP's, we weight the edges of the graph with posture similarity values. We scale posture similarity values to [0,1], where 1 is the specified threshold s_0. Transitions between GP's are

generated with motion blending to ensure smoothness. To optimize transitions, we compute the optimal blend length [7] for each pair of connected GP's. Finally, to avoid dead ends in the graph, we use Tarjan's algorithm to compute the largest strongly connected component (SCC) which will become the resulting GMG.

3.2.2 Visemes parameterization

We relate each phoneme with a viseme, which is represented by combination of blendshapes (shapes of the same mesh). To create a phoneme-viseme mapping we consider that multiple phonemes have similar mouth shapes when they are pronounced, therefore, they are linked to the same viseme. We use 15 categories (see Table 1).

Table 1: 15 phoneme categories. Each category maps to a single viseme. Symbols are codified with MRPA (Machine Readable Phonemic Alphabet).

/pau/	/r/	/k/, /g/, /ng/
/ae/, /ax/, /ah/, /aa/	/f/, /v/	/ch/, /sh/, /jh/
/ao/, /y/, /iy/, /ih/, /ay/, /aw/	/ow/, /oy/	/n/, /d/, /t/, /l/
/ey/, /eh/, /el/, /em/, /en/, /er/	/th/, /dh/	/s/, /z/, /zh/
/b/, /p/, /m/	/hh/	/w/, /uw/, /uh/

It is known that lip movements are linked to prosody [17]. Furthermore, the jaw lowers more in stressed syllables than in unstressed syllables [18]. Based on these statements, we propose a modification of visemes based on pitch accents strength. To that effect, we define a viseme blending space between high emphatic and low emphatic facial expressions, each one with appropriate jaw positions. For each vowel, three visemes are defined (see /ah/ and /aw/ phonemes examples in Figure 4): neutral, high emphatic and low emphatic.

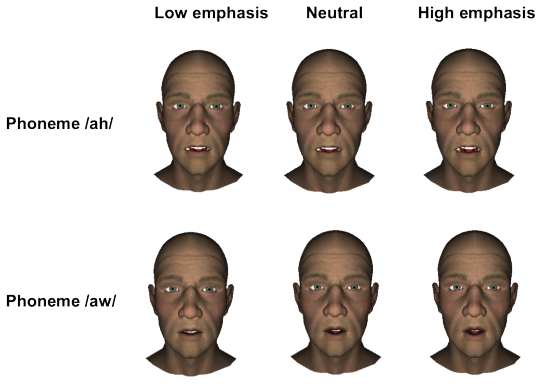

Fig. 4: Emphatic visemes for /ah/ and /aw/ phonemes.

3.3 Runtime stage

3.3.1 Pitch accents detection

Regarding pitch accent detection, we have developed a straightforward algorithm inspired by [5]. By pitch accent detection we mean detection of prominences in the speech stream. These prominences are potential candidates to be synchronized with gestures.

Taking a speech file as an input, we extract all the signal cycles with their associated information (amplitude, position, etc.). After selecting principal cycles, we extract voiced and unvoiced regions. Then, we extract and normalize pitch and intensity from voiced region nucleus (defined as maximum energy cycle inside the region) and compute the strength indicator as a sum of both parameters (see Figure 5). Finally, we have also detected pauses and we have rewarded voiced regions preceding a pause with extra strength indicator, as we observed that prosody tends to decrease in these situations causing undetected pitch accents.

Final pitch accents are detected according to the extracted strength indicators of the voiced region nucleus. Specifically, they are chosen depending on two tunable constraint parameters: strength indicator threshold and time difference threshold. Basically, the strength indicator threshold represents what percentage of the nucleus are pitch accents candidates (taking as a reference maximum strength indicator), and the time difference threshold defines how close pitch accents can be. If two pitch accent candidates are too close according to this parameter, we keep the one with the greater strength indicator. Finally, the pitch accents strength indicator is expressed in a [0,1] scale, taking as 1 the maximum strength indicator of the series.

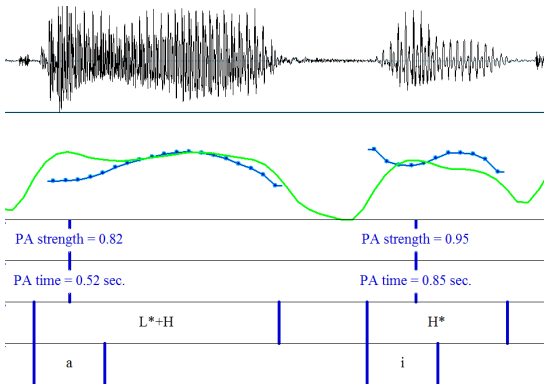

Fig. 5: Pitch accents detection. On top, there is the speech signal. Below, intensity (green) and pitch (blue) curves are displayed. In addition, pitch accent (PA) strength and time are shown. At the bottom, the intonation is represented (using the ToBI system [19]) with the affected vowel. The image was created thanks to the Praat software [20].

Furthermore, speech is analyzed to extract the phoneme transcription. So, we obtain a sequence of phonemes with

its type definition and timestamps. For each phoneme, initial time and final time are detected.

3.3.2 Body gesture synthesis

As we have explained, gesture synthesis is driven by pitch accents. Distances between consecutive pitch accent times define the duration of selected GPs, and pitch accents strength are related to GP's strength. We adopt FMDistance [21] to define GP strength using the reported parameters in [6]. Moreover, it is only computed for the stroke phase and it is normalized to [0,1]. Then, we iteratively evaluate each pitch accent and seek the most appropriate GP for each one. Gesture performance starts with a rest pose (which is also included in the motion graph as a node) and we use a breadth-first search algorithm to traverse the graph according to a proposed cost metric. Selected GP's are concatenated by motion blending to obtain a smooth motion stream. To finish the animation, the avatar returns to the rest pose.

a) Temporal alignment: Before computing the cost metric, candidate GPs (connected to the current node) are warped to temporally align them with the current pitch accent time. Our objective is to make the apex of the stroke coincide with the pitch accent time. However, it is known that gesture apexes are not exactly aligned with pitch accents [9]. In order to allow this de-synchronization, we compute an anticipation time for each pitch accent as a random value within a pre-defined window (from -0.03 to 0.22 seconds [6]) .

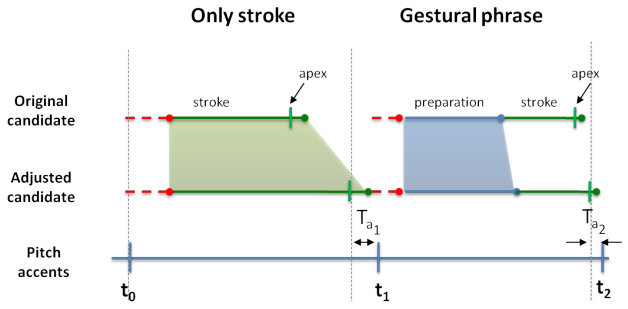

Fig. 6: Gesture alignment. Body gestures are aligned with pitch accents by modifying their length. The goal is to match stroke apexes with pitch accents times (t), taking into account anticipation times (T_a). We consider two cases: only stroke (on the left) and gestural phrase (on the right).

Furthermore, it is important to not modify strokes because they are the most significative part of gestures and emphasis relation in gesture selection is based on them. Hence, we manage two cases to align GPs with pitch accents times: 'only stroke' and 'gestural phrase'. 'Only stroke' means that the gestural phrase is formed by a unique stroke, in this case, stroke length will be modified. 'Gestural phrase' case means that GP has more phases besides the stroke, so, we modify phases which are not the stroke phase. Then, GP length (and its phases length) is computed by taking into account the mentioned cases, anticipation time and blending length (included as an edge weight in GMG) between the current node and the candidate one. Therefore, we obtain a w warping factor (original length divided by target length) for each candidate GP. In the 'only stroke' case, we consider the stroke length to compute w. On the other hand, we consider the sum of non-stroke phases lengths in the 'gestural phrase' case.

b) Gestural phrase selection: Our cost metric is based on: length similarity between a GP and the interval to fill (time cost), posture similarity between candidate GP and the previous one (smooth cost) and pitch accent strength-stroke strength relation (emphasis cost). As a result, we define our cost metric as

$$C(e(n_i, n_j), pa_k) = C_{smooth} + C_{emphasis} + C_{time} \quad (1)$$

where n_i is the previous GP, n_j is a candidate GP, and pa_k is the k-th pitch accent in speech stream. Smooth cost (C_{smooth}) is directly the posture similarity edge weight. Emphasis cost ($C_{emphasis}$) is the absolute difference between pitch accent strength indicator and gesture strength indicator. We recompute gesture strength indicator in the 'only stroke' time alignment case due to its duration, and consequently, its strength has changed. Time cost (C_{time}) is defined by

$$C_{time} = \begin{cases} w' & \text{if } \ min < w < max \\ p & \text{otherwise} \end{cases} \quad (2)$$

where w' is the normalized warping factor from [min,max] to [0,1]. We use min and max to not deteriorate motion quality by excessive changes on the original lengths. p is a penalty parameter that we use for penalizing GP's that exceed boundaries, avoiding their selection. Depending on the case of warping min and max take different values: 0.8 and 1.2 respectively for the 'gestural phrase' case, and 0.9 and 1.1 for the 'only stroke' case. Penalty parameter p is set to 10.

Once we have selected a gesture (the one with the minimum cost), this is concatenated with the previous one by linear motion blending. We use start-end blending scheme [7] and the blending length included in the edge weights of the graph.

3.3.3 Facial animation synthesis

We use phoneme transcription of the speech message to match phonemes with defined visemes. As usual, coarticulation between phonemes is generated by interpolating mesh points of visemes during initial and final times of phonemes. To include emphasis in facial expressions, we modify vowel

visemes by blending them with its emphatic visemes. This only occurs when a vowel phoneme matchs with a pitch accent. We relate pitch accent strength indicator with the amount of weight from neutral and high/low emphatic visemes. Pitch accent strength indicator is expressed in a 0 to 1 scale, so, we associate 0 values to low emphatic viseme, and 1 to high emphatic viseme as illustrated in Figure 7. So, pitch accent strength indicators that are lower than 0.5 will be represented by a combination of neutral and low emphatic viseme. Otherwise, neutral and high emphatic visemes will be used in the morphing process. In this way, we obtain the appropriate viseme according to speech intonation.

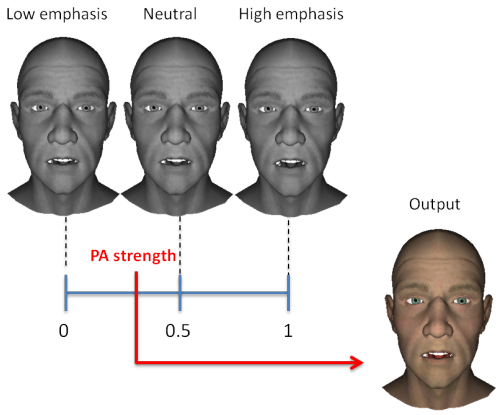

Fig. 7: Output viseme generation for pitch accents. Pitch accent (PA) strength is used to weight low emphatic, neutral and high emphatic visemes in the morphing process. In this example, higher emphatic visemes have more opened mouths.

4. Implementation

We have implemented the BodySpeech system as a plugin for the Unity3D game engine [22]. The Unity editor was used to create a visual interface that allows generating animations by selecting input speech audio files. In addition, the application uses Microsoft Speech API (SAPI) [23] to detect speech phonemes, and the Tagarela plugin [24] for facial morphing.

The application is able to parameterize the processes of gesture motion graph creation and viseme generation. Also, the process that synthesizes animations can be parameterized in order to modify emphasis, both for gesture and facial animations. This way,, animators can adjust output animations to satisfy plot requirements. The application is divided into three parts: New profile, Load profile and Player. New profile permits to generate a custom GMG (see Figure 8) and visemes; Load profile allows to select a saved profile; and Player lets to replay previous generated animations.

The process of generating new GMG's can be configured with the following parameters: joint weights (they are used in

Fig. 8: New graph screen. GMG can be parameterized by changing input databases, joint weights for posture similarity computation (dark blue box in the middle), threshold that defines the existance of transitions between GP's (slider on top right). Once the GMG is generated, graph information is displayed at the bottom of the screen to know graph capabilities.

posture similarity distance metric) and similarity threshold. Altering these parameters the GMG is modified. Moreover, the user can select one or several motion capture databases to be used as source of GP's for the GMG. This allows increasing the size of the GMG which in turn improves animation richness. Branching factor is displayed in the interface to lead animators know the richness of generated graphs. Also, visemes can be customized by changing the weights of former blendshapes.

Fig. 9: Synthesis screen. On the upper-left corner, there are the buttons to select an audio and generate the animations. At the bottom, there are the configurable pitch accent detection parameters, and sliders for adjusting gestural or facial animation emphasis.

As explained in Section 3.3.1, pitch accent detection can be parameterized by changing the strength indicator threshold and the time difference threshold. These two parameters

can be modified in the application affecting the frequency of detected pitch accents and gestures. A greater gesture frequency is perceived as a more emphatic animation. Moreover, emphasis of gesture and facial animations can be also be adjusted independently with two moving sliders. The gesture slider modifies the amount of strength that is added or subtracted to pitch accent strength (from -1 to +1). 0 denotes that the input pitch accent strength remains equal, positive values increase pitch accent strength value up to 1, while negative values decrease strength value down to -1. This permits the generation of more prominent gestures from a low emphatic speech, or contrarily, to relax gesticulation in a high emphatic speech. Similarly, facial animation emphasis is controlled by an analogous slider.

5. Conclusions and future work

In this paper, we have presented an automatic method to generate body gestures and facial animation according to speech input. Our animation system is based on motion graphs and lip sync techniques. Gesture animation stream is produced by concatenating gesture phrases aligned with pitch accents. Gestures are selected in order to maintain motion smoothness, preserve as many original motion clips as possible and obey emphasis relation with speech. Lip sync is generated following a standard algorithm. However, we relate speech strength with facial expressions to improve realism. Moreover, we have implemented a tool for animators that allows controlling the output animations via parameterization. A set of straightforward parameters are presented which permit a change in animation emphasis by adjusting pitch accents detection or emphasis relation between gestures/visemes with speech.

As future work, we plan to improve facial animation synthesis by studying the relationship between speech intonation and facial expressions. In addition, we plan to include independent head motion [25] and finger motion [26] to further increase realism of the overall animations.

Acknowledgements

This work was supported by the CENIT program number CEN-20101019, granted by the Ministry of Science and Innovation of Spain.

References

[1] D. McNeill, *Hand and Mind: What Gestures Reveal about Thought*. Chicago: University of Chicago Press, 1992.

[2] S. Levine, P. Krähenbühl, S. Thrun, and V. Koltun, "Gesture controllers," *ACM Trans. Graph.*, vol. 29, no. 4, pp. 124:1–124:11, July 2010.

[3] C.-C. Chiu and S. Marsella, "How to train your avatar: a data driven approach to gesture generation," in *Proceedings of the 10th international conference on Intelligent virtual agents*, ser. IVA'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 127–140.

[4] J. Lee, J. Chai, P. S. A. Reitsma, J. K. Hodgins, and N. S. Pollard, "Interactive control of avatars animated with human motion data," *ACM Trans. Graph.*, vol. 21, pp. 491–500, July 2002.

[5] O. Maeran, V. Piuri, and G. Storti Gajani, "Speech recognition through phoneme segmentation and neural classification," in *Instrumentation and Measurement Technology Conference, 1997. IMTC/97. Proceedings. Sensing, Processing, Networking., IEEE*, vol. 2, May, pp. 1215–1220 vol.2.

[6] A. Fernández-Baena, R. Montaño, M. Antonijoan, A. Roversi, D. Miralles, and F. Alías, "Gesture synthesis adapted to speech emphasis," *Speech Communication (Special Issue on Gesture and Speech in Interaction). In press.*

[7] J. Wang and B. Bodenheimer, "Synthesis and evaluation of linear motion transitions," *ACM Trans. Graph.*, vol. 27, pp. 1:1–1:15, March 2008. [Online]. Available: http://doi.acm.org/10.1145/1330511.1330512

[8] D. McNeill, "So you think gestures are nonverbal?" *Psychological Review*, vol. 92, no. 3, pp. 350–371, 1985.

[9] D. Loehr, "Gesture and intonation," Ph.D. dissertation, Georgetown University, 2004.

[10] T. Leonard and F. Cummins, "The temporal relation between beat gestures and speech," *Language and Cognitive Processes*, vol. 26, no. 10, pp. 1457–1471, 2011. [Online]. Available: http://www.tandfonline.com/doi/abs/10.1080/01690965.2010.500218

[11] J. Cassell, "Beat: The behavior expression animation toolkit." ACM Press, 2001, pp. 477–486.

[12] M. Stone, D. DeCarlo, I. Oh, C. Rodriguez, A. Stere, A. Lees, and C. Bregler, "Speaking with hands: creating animated conversational characters from recordings of human performance," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 506–513, Aug. 2004. [Online]. Available: http://doi.acm.org/10.1145/1015706.1015753

[13] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel, "Gesture modeling and animation based on a probabilistic re-creation of speaker style," *ACM Trans. Graph.*, vol. 27, no. 1, pp. 5:1–5:24, Mar. 2008. [Online]. Available: http://doi.acm.org/10.1145/1330511.1330516

[14] J. Terken, "Fundamental frequency and perceived prominence of accented syllables," *The Journal of the Acoustical Society of America*, vol. 89, no. 4, pp. 1768–1776, 1991.

[15] S. Levine, C. Theobalt, and V. Koltun, "Real-time prosody-driven synthesis of body language," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 172:1–172:10, Dec. 2009. [Online]. Available: http://doi.acm.org/10.1145/1618452.1618518

[16] A. Kendon, "Gesture and speech: two aspects of the process utterances," *Nonverbal Communication and Language*, pp. 207–227, 1980.

[17] E. Cvejic, J. Kim, and C. Davis, "It's all the same to me: Prosodic discrimination across speakers and face areas," in *Speech Prosody 2010-Fifth International Conference*, 2010.

[18] K. de Jong, M. Beckman, and J. Edwards, "The interplay between prosodic structure and coarticulation." *Lang Speech*, vol. 36 (Pt 2-3).

[19] K. Silverman, M. Beckman, J. Pierrehumbert, M. Ostendorf, C. Wightman, and J. Hirschberg, "TOBI: A standard scheme for labeling prosody," in *Proceedings of ICSLP-92*, Banff, October 1992, pp. 867–879.

[20] P. Boersma and D. Weenink, "Praat: doing phonetics by computer [computer program]. (v.5.2.29)," retrieved 12 July 2011 from http://www.praat.org/, 2011.

[21] K. Onuma, C. Faloutsos, and J. K. Hodgins, "FMDistance: A fast and effective distance function for motion capture data," in *Short Papers Proceedings of EUROGRAPHICS*, 2008.

[22] Unity, "Unity3d," 2013. [Online]. Available: http://www.unity3d.com/

[23] Microsoft, "Microsoft speech api," 2013. [Online]. Available: http://www.microsoft.com/en-us/download/details.aspx?id=10121

[24] R. Pegorari, "Tagarela - open source lip sync system for unity," 2013. [Online]. Available: http://rodrigopegorari.net/blog/?p=241

[25] C. Busso, Z. Deng, U. Neumann, and S. Narayanan, "Natural head motion synthesis driven by acoustic prosodic features: Virtual humans and social agents," *Comput. Animat. Virtual Worlds*, vol. 16, no. 3-4, pp. 283–290, July 2005. [Online]. Available: http://dx.doi.org/10.1002/cav.v16:3/4

[26] S. Jörg, J. Hodgins, and A. Safonova, "Data-driven finger motion synthesis for gesturing characters," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 189:1–189:7, Nov. 2012. [Online]. Available: http://doi.acm.org/10.1145/2366145.2366208