

A. Egges and Nadia Magnenat-Thalmann. Emotional Communicative Body Animation for Multiple Characters. V-Crowds'05, Lausanne, Switzerland. Pages 31-40. November 2005.

Emotional Communicative Body Animation for Multiple Characters

Arjan Egges, Nadia Magnenat-Thalmann

MIRALab - University of Geneva

24, Rue General-Dufour, 1211 Geneva, Switzerland

Telephone: +41 22 379 77 78, Fax: +41 22 379 77 80

Email: {egges,thalmann}@miralab.unige.ch

Abstract

Current body animation systems for Interactive Virtual Humans are mostly procedural or key-frame based. Although such methods provide for a high flexibility of the animation system, often it is not possible to create animations that are as realistic as animations obtained using a motion capture system. Simply using motion captured animation segments in stead of key-framed gestures is not a good solution, since virtual human animation systems also specify parameters of gesture that affect the style, such as for example expressing emotions or stressing a part of a speech sequence. In this paper, we describe an animation system that allows for the synthesis of realistic communicative body motions according to an emotional state, while still retaining the flexibility of procedural gesture synthesis systems. These motions are constructed as a blend of idle motions and gesture animations. Based on an animation specified for only a few joints, automatically and in real-time, the dependent joint motions are calculated. Realistic balance shifts adapted from motion capture data are generated on-the-fly, resulting in a fully controllable body animation, adaptable according to individual characteristics and directly playable on different characters at the same time.

Keywords

Computer Animation, Interactive Virtual Humans, Individuality

1 Introduction

Although virtual models of humans continue to improve both in real-time and non-real-time applications, controlling and animating them realistically still remains a difficult task. Techniques for capturing human motions and automatic adaptation of the obtained animations are maturing, but a gap still exists between animation engines and the systems that are controlling them, in particular Interactive Virtual Human (IVH) systems. In this paper, we will focus on two different types of motions:

Idle Motions An important aspect of an animation system is how to deal with a scenario where several animations are played sequentially for various actors. In nature there exists no motionless character, while in computer animation we often encounter cases where no planned actions, such as waiting for another actor finishing his/her part, is implemented as a stop/frozen animation. A flexible idle motion generator is required to provide for realistic motions even when no action is planned.

Nonverbal (Communicative) Body Motions Nonverbal body motions (gestures) are generally synthesized procedurally by an IVH system. As a result, the gestures often are not defined for all joints, but only for the arms, hands and head joints. However, when real humans are moving their arms, this affects other joints as well such as the spine and the shoulders. Generally, such kinds of motions are not defined in a high-level gesture specification. There is the need for a system that can automatically calculate these dependent joint movements in real-time and add them to the gesture animations.

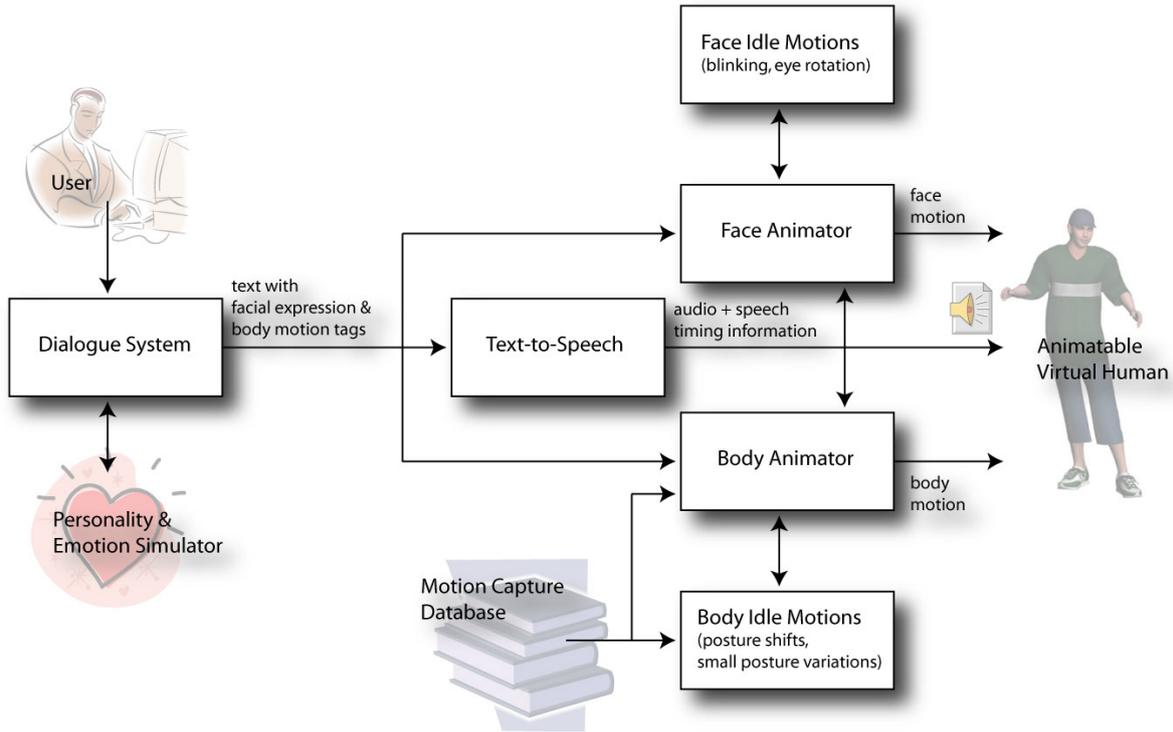


Figure 1: Overview of an Interactive Virtual Human animation system.

In our previous work [11, 12], we have developed an idle motion generator that constructs realistic idle animations from motion capture data, that still allows a high control from the animator or IVH control mechanism. We will give a short overview of this system in Section 3.2. Depending on which database of animations is used, different types of individuals can be portrayed through the body motions. In this paper, we provide for an extension of the idle motion engine that allows to generate different motions depending on the **emotional state** of an IVH. Additionally, we will describe a technique to automatically calculate dependent joints motions given a gesture animation specified for only a few joints (that is for example coming from a procedural gesture synthesis system). The resulting body motion is then integrated with a facial expression synthesizer, to produce realistic virtual human behaviour in synchrony with speech.

2 Background

Over the last years, quite some research has been done to develop systems that can simulate Interactive Virtual Humans (IVHs). Figure 1 shows an overview of how an animation system for simulating IVHs generally looks like. We will now discuss relevant research that has already been done in this area. One of the most well-known systems that can produce gesture animations from text, is BEAT [6]. BEAT allows animators to input typed text that they wish to be spoken by an animated human figure, and to obtain as output speech and behavioural characteristics. The MAX system, developed by Kopp and Wachsmuth [18], automatically generates gesture animations based on an XML specification of the output. In MAX, the gesture animations are generated procedurally (not from motion captured sequences). The work of Hartmann et al. [15] specifies a system that automatically generates hand and arm gestures from conversation transcripts using predefined key-frames. However, hand and arm gestures are not the only way in which the human body communicates. For example, Kendon [16] shows a hierarchy in the organization of movements such that the smaller limbs such as the fingers and hands engage in more frequent movements, while the trunk and lower limbs change relatively rarely. More specifically, posture shifts and other general body movements appear to mark the points of change between one major unit of communicative activity and another [28]. Cassell et al. [7] describe experiments to determine more precisely when posture shifts should take place during communication and they applied their technique to the REA agent [5], resulting in a virtual character being capable of performing posture shifts. All of the previously mentioned systems focus mainly on how to produce communicative body behaviour from language or a formal description of a communicative act and not so much on the **realism** of the resulting animations.

In order to provide for realistic animations, while still retaining (some) flexibility, motion captured data can be used and adapted to construct new animations. There are a broad number of algorithms for motion synthesis from motion data, although they are seldom directed to generating idle motions or gestures. Kovar et al. [19] proposed a technique called motion graphs for generating animations and transitions based on a motion database. Li et al. [22] divided the motion into textons modelled using linear dynamic system. Additionally, Kim et al. [17] propose a method that analyses sound and that can generate rhythmic motions based on the beats that are recognised in the audio. Pullen and Bregler [27] proposed to help the process of building key-frame animation by an automatic generation of the overall motion of the character based on a subset of joints animated by the user. Lee et al. [21] propose a motion synthesis method based on example motions that can be obtained through motion capture. Also relevant work has been done by Arikan et al. [3, 2] that define motion graphs based on an annotated motion database.

Recent work from Stone et al. [30] describes a system that uses motion capture data to produce new gesture animations. The system is based on communicative units that combine both speech and gestures. The existing combinations in the motion capture database are used to construct new animations from new utterances. This method does result in natural-looking animations, but in order to provide for a wide range of motions, a coherent performance from the motion captured person is required. Also, the style and shape of the motions are not directly controllable, contrary to procedural animation methods.

Especially when one desires to generate motions that reflect a certain style, emotion or individuality, a highly flexible animation engine is required that allows for a precise definition of how the movement should take place, while still retaining the motion realism that can be obtained using motion capture techniques. The EMOTE model [8] aims to control gesture motions using effort and shape parameters. As a result, gestures can be adapted to express emotional content or to stress a part of what is communicated. Currently no method exists that allows such expressive gestures, while still having a natural-looking final animation. Our method adapts gesture animation sequences *after* they have been fully constructed. Consequently, parameters such as effort and shape will still form a part of the final animation. However, a trade-off always has to be made between ensuring the exact execution of the effort and shape characteristics and having a realistic-looking motion.

It is generally assumed that an emotional state can be viewed as a set of dimensions. The number of these dimensions varies among different researchers. For example, Ekman [13] has identified six common expressions of emotion: fear, disgust, anger, sadness, surprise and joy, and in the OCC appraisal model [24], there are 22 emotions. Our system is based on an emotion representation called the **activation-evaluation space** [29], which defines emotions along two axes on a circle (see Figure 2), where the distance from the centre defines the power of the emotion. This emotion space allows for an easy mapping of different types of motions. Additionally, different discrete emotions can be placed on the disc [9], which provides for a possibility to link the activation-evaluation model with other multidimensional emotion models, such as the OCC emotions or Ekman's expressions.

The remainder of this paper is organised as follows. Section 3 will present an overview of the idle motion synthesizer and it will present an extension that allows to synthesize idle motions according to an emotional state. Section 4 will present our method to automatically create realistic gesture motions based on animations defined only for a few joints. Finally, we will show how the gesture animations and idle motions are integrated, as well as some results on different characters (Section 5).

3 Emotional Idle Motions

In this section, we will describe a system that can automatically produce realistic idle motions from segmented motion clips according to an emotional state. After we introduce the animation model that is used, we will give a overview of the idle motion engine. For a more detailed description of the idle motion generator, please see our previous work [11]. In Section 3.3 we will then present an extension of this system that allows for emotional idle motions.

3.1 Animation Model

There exist many techniques for animating virtual characters. Two very commonly used techniques are:

- **Key-frames:** an animation is constructed from a set of key-frames (manually designed by an animator or generated automatically) by using interpolation techniques. Although this method results in very flexible animations, the realism of the animations is low, unless a lot of time is invested.
- **Pre-recorded animations:** an animation is recorded using a motion capture/tracking system such as Vicon or MotionStar. The animation realism is high, but the resulting animation is usually not very flexible,

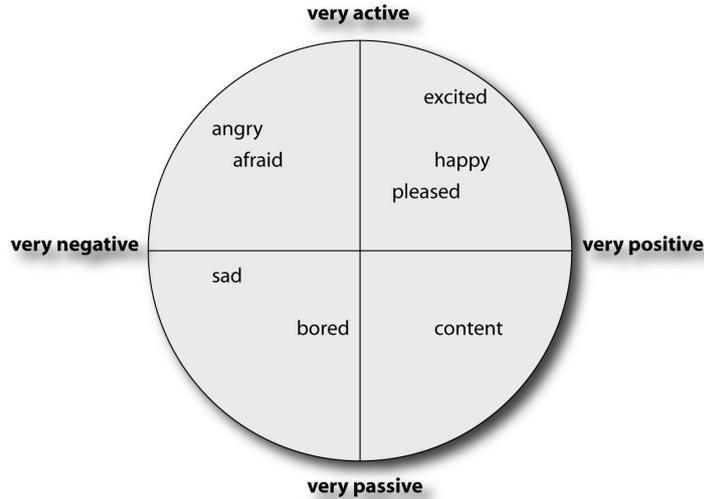


Figure 2: Activation-evaluation emotion disc.

although methods have been developed to overcome part of this problem [4].

A method like Principal Component Analysis (PCA) can determine dependencies between variables in a data set. The result of PCA is a matrix (constructed of a set of eigenvectors) that converts a set of partially dependent variables into another set of variables that have a maximum independency. The PC variables are ordered corresponding to their occurrence in the dataset. Low PC indices indicate a high occurrence in the dataset; higher PC indices indicate a lower occurrence in the dataset. As such, PCA is also used to reduce the dimension of a set of variables, by removing the higher PC indices from the variable set. We will use the results of the PCA later on for synthesizing the dependent joint motions (see Section 4). For our analysis, we perform the PCA on a subset of H-Anim joints. In order to do that, we need to convert each frame of the animation sequences in the data set into an N -dimensional vector. For representing rotations, we use the exponential map representation [14]. In this representation, a rotation can be represented by a vector $r \in \mathbb{R}^3$, as a rotation with angle $\|r\|$ around axis r .

The exponential map representation of a rotation is very useful for motion interpolation [25, 1], because it allows to perform linear operations on rotations. In our case the linearity of the exponential map representation is crucial since the PCA only works in the linear domain. Any rotation matrix can be written in the exponential map representation, and any exponential map representation (modulo 2π) is a rotation. Grassia [14] provides an extensive overview of the advantages and disadvantages of various representations of rotations, including the exponential map.

Using the exponential map representation for a joint rotation, a posture consisting of m joint rotations and a global root translation can be represented by a vector $v \in \mathbb{R}^{3m+3}$. In our case, one posture/key-frame is represented by 25 joint rotations and one root joint translation, resulting in a vector of dimension 78. We have applied a PCA on a large set of motion captured postures, resulting in a PC space of equal dimension.

3.2 Idle Motion Synthesis

We have separately recorded the motions of ten people of both genders while they were in a conversation. This provides us with motion data of both gestures and idle motions. In the recorded data, we have observed three common types of idle behaviour:

Posture shifts This kind of idle behaviour concerns the shifting from one resting posture to another one. For example, shifting balance while standing, or go to a different lying or sitting position.

Continuous small posture variations Because of breathing, maintaining equilibrium, and so on, the human body constantly makes small movements. When such movements are lacking in virtual characters, they look significantly less lively.

Supplemental idle motions These kinds of motions generally concern interacting with ones own body, for example touching of the face or hair, or putting a hand in a pocket.

3.2.1 Balance Shifting

Humans need to change posture once in a while due to factors such as fatigue. Between these posture changes, he/she is in a resting posture. We can identify different categories of resting postures, such as in the case of standing: balance on the left foot, balance on the right foot or rest on both feet. Given a recording of someone standing, we can extract the animation segments that form the transitions between each of these categories¹. These animation segments together form a **database** that is used to synthesize balancing animations. In order for the database to be usable, at least one animation is needed for every possible category transition. However, more than one animation for each transition is better, since this creates more variation in the motions later on. In order to generate new animations, recorded clips from the database are blended and modified to ensure a smooth transition.

Once the transitions between the different postures have been calculated, the creation of new animations consists of simply requesting the correct key frame from the database during the animation. This means that the database can be used to control many different virtual humans at the same time. For each virtual human, a different motion program is defined that describes the sequence of animation segments that are to be played. This motion program does not contain any real motion data but only references to transitions in the database. Therefore it can be constructed and updated on-the-fly.

3.2.2 Small Posture Variations

Apart from the balance shifting postures, small variations in posture also greatly improve the realism of animations. Due to factors such as breathing, small muscle contractions etc., humans can never maintain the exact same posture. As a basis for the synthesis of these small posture variations, we use the Principal Component representation for each key-frame. Since the variations apply to the Principal Components and not directly to the joint parameters, this method generates randomised variations that still take into account the dependencies between joints. Additionally, because the PCs represent dependencies between variables in the data, the PCs are variables that have *maximum independency*. As such, we can treat them *separately* for generating posture variations. The variations can be generated either by applying a Perlin noise function [26] on the PCs or by applying the method that is described in our previous work [11]. Small posture variations are very important when the character is well visible on the screen. For crowds however, characters that are further away do not need such small variations since they will not be visible anyway. In the case of many virtual humans in one scene, it suffices to synthesize these variations for the characters that are close to the camera.

3.3 Emotional Idle Motions

In the evaluation-activation space, an emotional state e is defined as a 2-dimensional vector:

$$[e_e, e_a], \text{ where } -1 \leq e_e, e_a \leq 1 \text{ and } \sqrt{e_e^2 + e_a^2} \leq 1 \quad (1)$$

When using a discrete list of n different emotion dimensions (for example based on OCC), a mapping function $f : \mathbb{R}^n \rightarrow \mathbb{R}^2$ that respects the conditions in Equation 1 has to be defined. In this way, the emotional state representation that is used in this paper can be linked with frameworks of emotion simulation such as for example presented in our previous work [10].

3.3.1 Emotional Balance Shifting

The extension of the idle motion engine for now mainly focuses on the balance shift synthesizer. The animations that are in the animation database are extended with additional emotional information. For each animation segment, we define the change of emotional content (if any) by specifying for both the start and end points of the animation a 2-dimensional interval on the activation-evaluation circle. Figure 3 shows some examples of possible intervals and related postures on the activation-evaluation circle.

So, given an emotional state $[e_e, e_a]$, the idle motion synthesizer automatically selects animations that have a target interval including this point in the emotion space. In order to make sure that it is always possible to make a balance shift regardless of the emotional content, a set of neutral posture shifts is added as well, so that when no suitable target interval can be selected, a posture shift is still possible.

¹In the current configuration this segmentation is done manually, however automatic segmentation methods also exist [23].

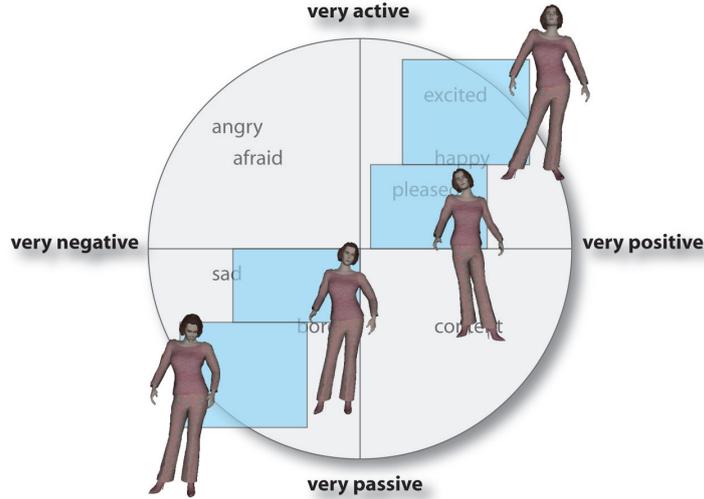


Figure 3: Different intervals of emotional states together with example postures.

3.3.2 Adapting the pause length

The balance motion synthesizer together with facial expressions can portray postures and shifts that change according to evaluation (positive or negative), together with the activation level. An additional option that we have implemented is the automatic adaptation of pause length according to the activation level. Higher activation level will result in shorter pauses (and thus more shifts). In our system the length of a pause is determined using a minimum length p_m and a maximum offset p_o . A random value between p_m and $p_m + p_o$ is then chosen as the final pause length. In order to adapt this value according to the emotional state, the value of p_o is replaced by p_o' , which is calculated as follows:

$$p_o' = (\alpha - e_a) \cdot \beta \cdot p_o, \text{ and } \alpha \geq 1, \beta \geq 0 \quad (2)$$

where α and β are parameters that define how the offset length adaptation should be applied. In our system these values are set by default to $\alpha = 1$ and $\beta = 1$. The application of the length adaptation can be dynamically switched on and off, so that there is no interference when pauses are required to have specific lengths (for example during speech).

4 Natural Gestures

As discussed in Section 2, body gesture synthesis systems often generate gestures that are defined as specific arm movements coming from a more conceptual representation of gesture. Examples are: “raise left arm”, “point at an object”, and so on. Translating such higher level specifications of gestures into animations often results in motions that look mechanic, since the motions are only defined for a few joints, whereas in motion captured animations, each joint motion also has an influence on other joints. For example, by moving the head from left to right, some shoulder and spine movements normally occur as well. However, motion captured animations generally do not provide for the flexibility that is required by gesture synthesis systems.

Such systems would greatly benefit from a method that can automatically and in real-time calculate believable movements for the joints that are dependent on the gesture. Methods that can calculate dependent joint motions already exist, see for example the research done by Pullen and Bregler [27]. In their work, they adapt key-framed motions with motion captured data, depending on the specification of which degrees of freedom are to be used as the basis for comparison with motion capture data. In this paper, we will present a novel method that uses the Principal Components to create more natural looking motions. Our method is not as general as the previously discussed work, but it works very well within the upper body gesture domain. Furthermore, it is a very simple method and therefore suited for real-time applications.

The Principal Components are ordered in such a way that lower PC indices indicate high occurrence in the data and higher PC indices indicate low occurrence in the data. This allows for example to compress animations by only retaining the lower PC indices. Animations that are close to the ones that are in the database that was used for the PCA, will have higher PC indices that are mostly zero (see Figure 4) for an example. An animation that is very different from what is in the database, will have more noise in the higher PC indices to compensate

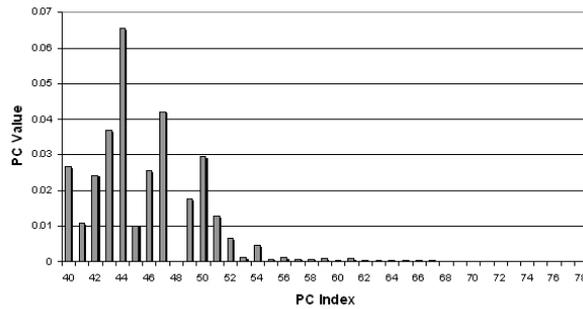


Figure 4: (Absolute) PC values of a posture extracted from a motion captured animation sequence.

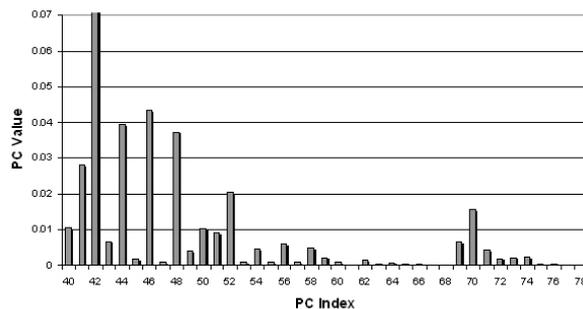


Figure 5: (Absolute) PC values of a posture modelled by hand for a few joints.

for the difference (see Figure 5). If one assumes that the database that is used for the PCA is *representative* for general motions that are expressed by humans during communication, then the higher PC indices represent the part of the animation that is ‘unnatural’ (or: not frequently occurring in the animation database). When we remove these higher PC indices or apply a scaling filter (such as the one displayed in Figure 6), this generates an error in the final animation. However, since the scaling filter removes the ‘unnatural’ part of the animation, the result is a motion that actually contains the movements of dependent joints. By varying the PC index where the scaling filter starts, one can define how close the resulting animation should be to the original key-framed animation.

To calculate the motions of dependent joints, only a scaling function has to be applied. Therefore this method is very well suited for real-time applications. A disadvantage is that when applying the scaling function onto the global PC vectors, translation problems can occur. In order to eliminate these translation artefacts, we have also performed a PCA on the upper body joints only (which does not contain the root joint translation). The scaling filter is then only applied on the upper body PC vector. This solution works very well since in our case, the dependent joint movements are calculated for upper body gestures only, whereas the rest of the body is animated using the idle motion engine. Figure 7 shows some examples of original frames versus frames where the PC scaling filter was applied.

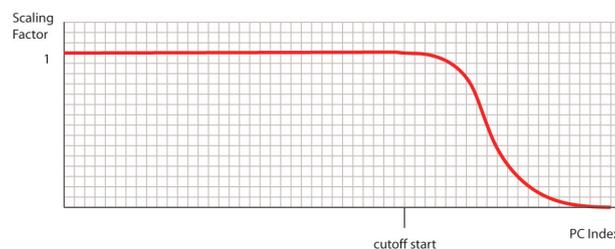


Figure 6: An example of a scaling filter that can be applied to the PC vector representation of a posture.



Figure 7: Two examples of key frame postures designed for a few joints and the same postures after application of the Principal Component scaling filter.

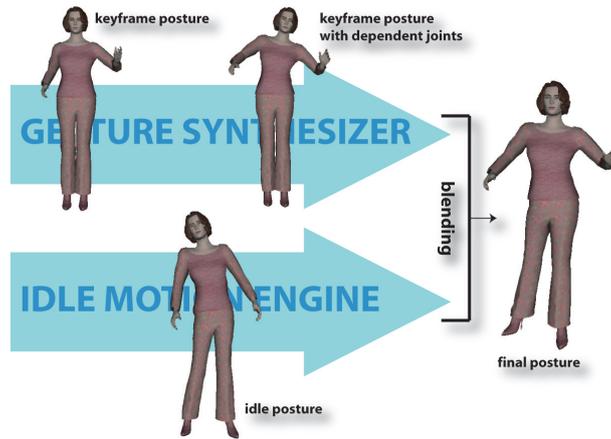


Figure 8: Integration of gesture animations, dependent joint motion synthesis and idle motion synthesis.

5 Integration and Results

In order to integrate the emotional idle motions and the gesture animations, we use the blending library that was developed in our earlier work [12]. This library allows us to perform weighted animation blending operations in the exponential map space. Additionally, a set of modifiers is provided that allows scaling, flipping and stretching of animations, among others. Figure 8 shows the general process of the animation synthesis and blending. The idle motion engine is running continuously, therefore providing the IVH with continuous idle motions. Gesture animations are adapted so that the dependent joint movements are also included, and are blended in on-the-fly.

The blending parameters such as weight are currently set as default values. For the lower body, 100% of the idle motion is used, whereas for the upper body 75% of the gesture animation is used and 25% of the idle motion. Each gesture animation is designed in such a way that it starts and ends in the H-Anim neutral posture. A blending fade-in and fade-out of 500 ms is also applied on the gesture animation, in order to avoid unnatural transitions. In our case, these values and percentages worked well for most of the gestures. However, some animations might need different values if some parts of the animation are very important and should not be changed. The PC scaling filter is applied on the upper body only and starts at PC index 20 (of a total of 48). The final body animation is then synchronized with a speech signal and a face animation [20] (see Figure 9).

6 Conclusions and Future Work

We have presented an idle motion engine that can produce realistic looking animations according to an emotional state, based on motion captured animation segments. Additionally, we have shown a method to automatically determine the movements of dependent joints, given a key frame animation specified for only a few joints. Our system then blends both the animations, resulting in a final animation that can be synchronised with facial animation and speech. While our method improves the realism of animations with communicative and emotional content, it still retains the flexibility that is required for IVH systems. Because of the methods that are used, all of the operations are performed in real-time. The system works very well for generic gesture animations, although



Figure 9: Some frames of gesture and idle motion sequences played on different 3D models, synchronised with facial animation and speech.

there is still room for improvements.

Since the system currently uses default parameters for the blending procedure, some important parts of the gesture animations might be affected when blended with the idle motions. A solution could be that in the gesture specification, a tag is included that indicates if a part of a gesture (or a set of joints) should not be affected by movements from other animations (such as idle motions). Our future work will focus on how these blending parameters can be easily integrated with high-level gesture synthesizers, while still ensuring that controlling the gesture sequence remains simple. Another limitation of our method is that the dependent joint calculation only works if a representative database is used.

Finally, if one desires to simulate a lot of different emotional states, this also means that a large database of motions is required. In order to further improve the flexibility of the system, we will investigate if it is possible to apply interpolation techniques between the motions, so that not all emotional states need to be recorded. For example, by interpolating between *extremely happy* and *neutral* motions, the *medium happy* sequences could be constructed automatically.

7 Acknowledgements

This work has been developed through the support of the HUMAINE Network of Excellence (Human-Machine Interaction Network on Emotion, IST-507422, <http://www.emotion-research.net>) funded under the Sixth Framework Programme and by the OFES.

References

- [1] Marc Alexa. Linear combination of transformations. In *SIGGRAPH 2002*, pages 380–387, 2002.
- [2] O. Arikan and D. Forsyth. Interactive motion generation from examples. In *Proceedings of ACM SIGGRAPH 2002*, 2002.
- [3] Okan Arikan, David A. Forsyth, and James F. O’Brien. Motion synthesis from annotations. *ACM Transactions on Graphics*, 22(3):392–401, 2003.
- [4] Armin Bruderlin and Lance Williams. Motion signal processing. *Computer Graphics*, 29(Annual Conference Series):97–104, 1995.
- [5] J. Cassell, T. Bickmore, M. Billingham, L. Campbell, K. Chang, Vilhjálmsson, H., and H. Yan. Embodiment in conversational interfaces: Rea. In *Proceedings of the CHI’99 Conference*, pages 520–527, 1999.
- [6] J. Cassell, H. Vilhjálmsson, and T. Bickmore. BEAT: the Behavior Expression Animation Toolkit. In *Proceedings of SIGGRAPH*, pages 477–486, 2001.
- [7] Justine Cassell, Yukiko I. Nakano, Timothy W. Bickmore, Candace L. Sidner, and Charles Rich. Non-verbal cues for discourse structure. In *Proceedings Association for Computational Linguistics Annual Conference (ACL)*, pages 106–115, July 2001.

- [8] D. Chi, M. Costa, L. Zhao, and N. Badler. The emote model for effort and shape. In *SIGGRAPH 2000*, pages 173–182, July 2000.
- [9] R. Cowie, E. Douglas-Cowie, S. Savvidou, E. McMahon, M. Sawey, and M. Schröder. Feeltrace: An instrument for recording perceived emotion in real time. In *ISCA Workshop on Speech and Emotion*, pages 19–24, Northern Ireland, 2000.
- [10] A. Egges, S. Kshirsagar, and N. Magnenat-Thalmann. Generic personality and emotion simulation for conversational agents. *Computer Animation and Virtual Worlds*, 15(1):1–13, 2004.
- [11] A. Egges, T. Molet, and N. Magnenat-Thalmann. Personalised real-time idle motion synthesis. In *Pacific Graphics 2004*, pages 121–130, 2004.
- [12] A. Egges, R. Visser, and N. Magnenat-Thalmann. Example-based idle motion synthesis in a real-time application. In *CAPTECH Workshop*, pages 13–19, 2004.
- [13] P. Ekman. *Emotion in the human face*. Cambridge University Press, New York, 1982.
- [14] F. Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.
- [15] Björn Hartmann, Maurizio Mancini, and Catherine Pelachaud. Formational parameters and adaptive prototype instantiation for mpeg-4 compliant gesture synthesis. In *Computer Animation 2002*, pages 111–119, 2002.
- [16] Adam Kendon. Some relationships between body motion and speech: an analysis of one example. In Aron Wolfe Siegman and Benjamin Pope, editors, *Studies in Dyadic Communication*, pages 177–210. New York: Pergamon, 1972.
- [17] T. H. Kim, S. I. Park, and S. Y. Shin. Rhythmic-motion synthesis based on motion-beat analysis. *ACM Transactions on Graphics*, 22(3):392–401, 2003.
- [18] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds*, 15(1):39–52, 2004.
- [19] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proc. SIGGRAPH 2002*, 2002.
- [20] S. Kshirsagar, S. Garchery, and N. Magnenat-Thalmann. Feature point based mesh deformation applied to MPEG-4 facial animation. In *Proceedings Deform2000*, pages 23–34, Geneva, Switzerland, November 2000.
- [21] Jehee Lee, Jinxiang Chai, Paul Reitsma, Jessica K Hodgins, and Nancy Pollard. Interactive control of avatars animated with human motion data. In *Proceedings of SIGGRAPH 2002*, July 2002.
- [22] Y. Li, T. Wang, and H. Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. In *Proc. SIGGRAPH 2002*, 2002.
- [23] Meinard Mueller, Tido Roeder, and Michael Clausen. Efficient content-based retrieval of motion capture data. In *Proceedings SIGGRAPH 2005*, 2005.
- [24] Andrew Ortony, Gerald L. Clore, and Allan Collins. *The Cognitive Structure of Emotions*. Cambridge University Press, 1988.
- [25] F. C. Park and Bahram Ravani. Smooth invariant interpolation of rotations. *ACM Transactions on Graphics*, 16(3):277–295, July 1997.
- [26] Ken Perlin. An image synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 287–296. ACM Press, 1985.
- [27] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. In *Proc. SIGGRAPH 2002*, 2002.
- [28] A. Schefflen. *Communicational structure*. Bloomington: Indiana University Press, 1973.
- [29] H. Schlosberg. A scale for judgement of facial expressions. *Journal of Experimental Psychology*, 29:497–510, 1954.
- [30] Matthew Stone, Doug DeCarlo, Insuk Oh, Christian Rodriguez, Adrian Stere, Alyssa Lees, and Christoph Bregler. Speaking with hands: Creating animated conversational characters from recordings of human performance. In *SIGGRAPH 2004*, pages 506–513, 2004.