A. Egges, R. Visser, N. Magnenat-Thalmann. Example-Based Idle Motion Synthesis in a Real-Time Application. CAPTECH Workshop, Zermatt, Switzerland, pp. 13-19. December 2004.

Example-Based Idle Motions in a Real-Time Application

Arjan Egges Robert Visser
Nadia Magnenat-Thalmann

MIRALab - University of Geneva
Switzerland
{egges,visser,thalmann}@miralab.unige.ch

Abstract

In this paper, we describe an animation synthesizer that allows generating two layers of subtle motions: small posture variations and personalised change of balance. Such a motion generator is needed in many cases when one attempts to create an animation sequence out of a set of existing clips, in order to avoid unnatural-looking transitions or frozen waiting-states between clips. The motion synthesizer forms a part of a real-time blending engine that allows to smoothly mix the idle motions with other animation clips. Personalised animations can be obtained by using different animation databases for different characters.

Keywords: motion analysis, idle motion synthesis, personality, evaluation

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. 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. We identify many situations where a flexible idle motion generator can help: from synchronisation of speech/body animation duration, to dynamic creation of stand still variations in between two active plays. Additionally, as every person has a unique way of moving and standing, an individualised motor control system is required to transfer this property to Virtual Human motions.

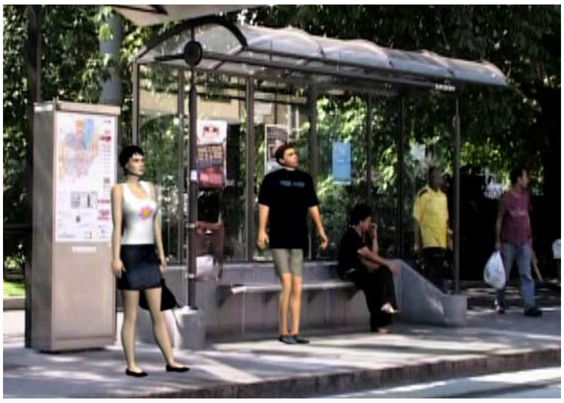

Figure 1: Virtual Humans waiting at a tram stop in Geneva.

From now on, we will call the motions that occur in waiting/idle states between animation clips **idle motions**. These motions include changing balance because of fatigue, small variations in body posture caused by small muscle contractions or eye blinking. Apart from generating random movements for separate joints, another possibility is to take captured animation files as a basis for generating idle motions. This results in idle motions that affect all joints, but unfortunately, they are very repetitive and in-flexible. Secondly, this approach generates transition problems between the animation clips.

In this paper, we will discuss our techniques to realistically synthesize idle motions using motion captured example animations. We will show how the idle motions can be blended with other animations, such as gestures. Finally, we will discuss the results of a user evaluation of the system that we have performed.

2 Background

There are several research endeavours concerning the development of a gesture synthesizer. The BEAT project [1] 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. Another project that concerns gesture synthesis is REA [2]. REA (or: the Real Estate Agent) is based on similar principles as the BEAT system. Finally, we would like to mention the MAX system, developed by Kopp and Wachsmuth [3]. This system generates automatically gesture animations based on an XML specification of the output. Most of these systems only focus on gestures performed by the hand and arms. However, there are many indications that the whole body is used during communications. For example, Kendon [4] 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 [5]. Also, posture shifts are used as a means to communicate *turn-taking*[1] during dialogues [6]. 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 Embodied Conversational Agent [2], resulting in a virtual character being capable of performing posture shifts. The posture shifting animations themselves often lack realism, since they are generated from a limited set of postures and are thus repetitive. However, the realism can be improved by using motion captured data.

There is a broad number of algorithms for motion synthesis from motion data, although they are seldom directed to addressing the idle-motion concerns. Kovar et al. [8] proposed a technique called motion graphs for generating animations and transitions based on a motion database. Li et al. [9] divided the motion into textons modelled using linear dynamic system. Additionally, Kim et al. [10] propose a method that analyses sound and that can generate rhythmic motions based on the beats that are recognised in the audio. These techniques are best suited for motions consisting of rhythmic patterns such as disco dance. Pullen and Bregler [11] 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. [12] 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. [13, 14] that define motion graphs based on an annotated motion database.

Another animation method to increase realism of virtual actors is the application of a *noise function* on one or more joints to ensure that the virtual body is never really static when no animation is playing. This method is described by Perlin [15], based on the application of Perlin-noise [16] on elbow, neck and pelvis joints. This method generates quite realistic animation noise, while the noise is applied onto a few joints. However, real human posture variations affect all joints, which cannot easily be solved by using noise functions on all joints, because of dependencies between joints.

In our previous work [17], we describe a system that allows to automatically generate non-repetitive idle motions, taking into account the previously mentioned concerns. In Section 3 we will give an overview of the idle motion synthesizer. Then, we will show how the idle motions can be blended into a real-time application in Section 4. Finally, Section 5 shows the results and discusses a user evaluation of the system.

[1] They are called *situational shifts* by Blom and Gumperz [6].

3 Multi-level Idle Motions

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. Based on the recorded data and the related work discussed in the previous section, we define 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.

In our system [17], we focus on the first two types of variations. The supplemental idle motions are currently generated by using prerecorded sequences. This is not a very flexible solution and it is part of our future work to extend the system so that it can also generate these kinds of motions automatically.

3.1 Posture Changing

Every person has a distinctive way of standing, sitting, lying and moving. Our goal is to capture (at least some of) this personalised way of moving. In this section we will describe an approach that allows creating an animation database from recorded clips. From this database, we can generate fully automatically new realistic animations. This results in a Virtual Human that mimics the postures and the way of moving of the person that was recorded. As an example to illustrate our techniques, we use recordings of people standing.

When a human is standing, he/she needs to change posture once in a while due to factors

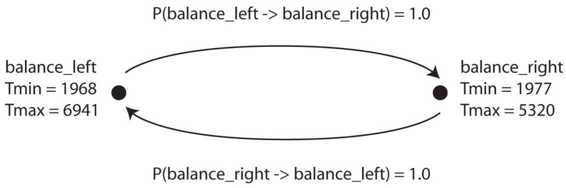

Figure 2: An example of a category transition graph. For both the categories, a minimum and maximum duration (in milliseconds) is determined from the data.

such as fatigue. Between these posture changes, he/she is in a resting posture. We can identify different categories of resting postures, such as: balance on the left foot, balance on the right foot or rest on both feet. As such, given a recording of someone standing, we can extract the animation segments that form the transitions between each of these categories (see Figure 2 for an example of possible transitions). 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.

The animation database is constructed from one or more recorded animations of a person. We feel that not only the basic postures of someone, but also the way a person changes posture, is unique for everyone. This feeling is strengthened by our findings that, in general, people were able to recognise whose animations were at the basis of the animations generated by the method described before. According to this subjective analysis, our technique allows grasping the way someone stands and moves. We have constructed personalised animation databases for several people, which allows one to choose which person a virtual human should mimic.

3.2 Continuous Small Variations in Posture

Apart from the balance shifting postures that were discussed in the previous section, small posture variations also greatly improve the realism of animation. 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 [17]. 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.

One method to generate variations, is to use Perlin Noise [16] functions for each—or a subset—of the Principal Components. Another method presented in earlier work [17] generates these small variations based on the shape of the curves in the motion capture data. This method applies a statistical model to generate similar (randomised) curves. This approach also allows keeping certain existing tendencies in the variations that are specific to certain persons (such as typical head movements, etc.). An additional advantage over using Perlin noise, is that this approach is fully automatic and it avoids the need to define frequencies and amplitudes to generate noise, although these parameters could eventually be determined automatically by analysing the signal. In our method, we analyse animation segments that do not contain any balance shifts or other motions other than the small variations. The results of the analysis will then be used to generate similar variations on top of other postures.

4 Blending

The blending part of the application is responsible for combining all, possibly simultanious animations into one. First we will describe the blending library, then later how it is used in the blending application.

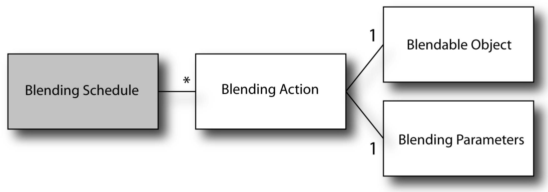

Figure 3: The main structure of the blending library.

4.1 The Blending Library

For the blending a generic blending library is being used. The main class in this library is the blending schedule, that contains a list of blending actions. Each of these blending actions is a combination of a blendable object, and a blending parameters object , see Figure 3.

A blendable object is an object that defines for a certain duration, on each time instance a vector of components.

The blending parameters object specifies for each component in the blendable object for each time instance two properties: a weight and a scale. The scale specifies a factor by which the component is multiplied. The weight specifies the importance of a component in the blending result.

The blending library supports three blend modes. The first is the weighted average. Of all actions that are in this mode the weighted avarage is taken. An action in the second blend mode is also taken in this weighted avarage, but its weight is determined by the weights of the other actions, by formula 1.

$$max(0, 1 - \Sigma otherweights) \qquad (1)$$

The third blend mode is the normal addition. An action in this mode is simply added up to the other animation.

Beside blending the blending engine also supports modification of blendable objects. A modifier can be applied to a blendable object, resulting in a new blendable object. To this result a new modifier can be applied, and so a whole stack of modifiers can be applied to a blendable object. Using such a stack, modifiers can be applied in arbitrary order, their parameters can be changed and they can be removed again if desired.

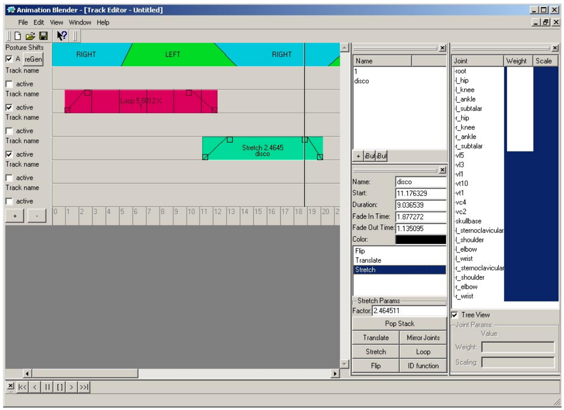

Figure 4: Screenshot of the blending application.

4.2 *The Animation Blending Application*

For the blending of the joint angles, multiple ways of interpolation are possible. Two commonly used interpolation methods are SLERP interpolation of quaternions [18] and the exponential map [19]. The latter, which we use, gives us a linear representation of joint angles that can be blended by simply taking a weighted average.

The blending application has a track based user interface. The tracks are time lines on which animations can be placed. Once placed, these animations can be dragged around through time, their parameters can be changed and modifiers can be added and removed. Examples of modifiers on animations are stretching in time, translation in space and looping. In addition to the normal tracks, there is an idle motion track that describes the progress of the posture changing. While the sequence and its animations are modified the result is shown on a virtual human. A screenshot of the application is given in Figure 4

The blending parameters of animations are used as follows. For each joint a scaling and a weight can be specified. With the scaling, motion can be exagerrated, or lessened if desired. With the weight the influence of the joint in the avarage that is taken can be specified. Weights are also used to make transitions between animations go smoothly. This is done by gradually raising and lowering the weights at the start and end phases of the animation.

The weight of the posture changing motion is determined by the weights of the other animations by formula 1. This way the posture shifting only happens when the total weight of the other animations is below 1. The variation motion is simply added up to the other animations.

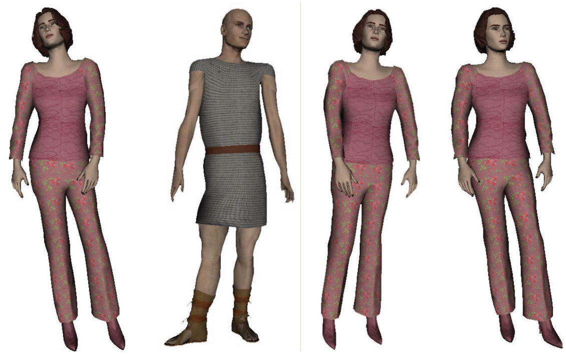

Figure 5: Some example postures from different databases.

5 Results and Evaluation

The techniques that are described in this paper allow to create a system that, based on a personalised database of animations and postures, can realistically reconstruct idle motions that are coherent with the animations and postures in the database (see Figure 5).

For evaluation purposes, we have created personalised databases for different people. We then performed a user evaluation (15 subjects) of the idle motion synthesizer. The test subjects were shown different movies of virtual character playing either a recorded idle motion sequence or an idle motion sequence that was synthesized using our technique. The movies contained recorded and generated idle motions of 8 different people. The subjects were then asked three questions:

1. what is the gender of the person that performs the animation?
2. who is the person that performs the animation?
3. is the animation recorded using motion capture or automatically generated?

The evaluation results are shown in Figure 6. The gender recognition is a bit better for the recorded sequences (about 7% higher than the recognition percentage of the automatically generated idle motions). Recognising the person

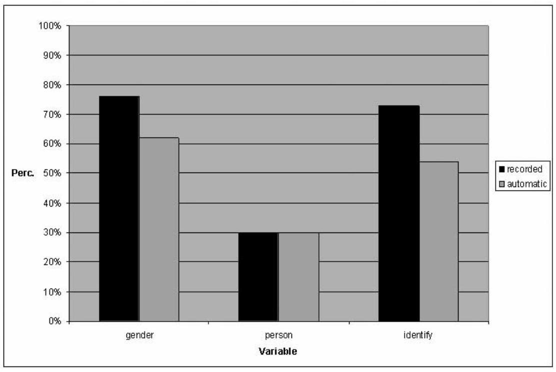

Figure 6: Overview of the user evaluation results.

from the animation is quite difficult—in about one-third of the cases success—but it doesn't make any difference whether or not the motions are recorded or synthesized. This leads us to believe that our technique succesfully captures (a part of) the individuality in the motions. Finally, the recorded motions were correctly identified as recordings in 73% or the cases, wheras the automatically generated motions were correctly identified in 54% of the cases. This is an indication that although sometimes the automatically generated motions are being observed as fake, they still look like recorded motions in almost 50% of the cases[2].

6 Conclusions and Future Work

We have developed a flexible idle motion engine as a part of a real-time animation system that requires realistic motions of virtual humans during waiting/idle states. The motions are generated on the fly from a database of personalised animations and postures, obtained by motion capture examples. The motions are then blended with other animations. Furthermore we have described the results of a user evaluation of the animation quality.

In the near future, we will work on further improving the quality of the idle motions. First of all, we aim at enhancing the link between the small posture variation engine and the balance-shifting engine, in order to develop additional features, such as 'stabilising variations' just after the transition from one balance to another. Additionally, we would like to explore other possibilities that the Principal Components offer for improving the realism of other motions, such as body gestures. Finally, we would like to link the idle motion synthesizer with descriptive models of personality and emotion from psychology.

[2] This percentage is probably higher, since more than 25% of the recorded motions were incorrectly classified as being automatically generated.

Acknowledgements

This work has been developed through the support of the AIM@Shape Network of Excellence (Advanced and Innovative Models And Tools for the development of Semantic-based systems for Handling, Acquiring, and Processing Knowledge Embedded in multidimensional digital objects, IST-506766, http://www.aimatshape.net) funded under the Sixth Framework Programme and by the OFES.

References

[1] J. Cassell, H. Vilhjálmsson, and T. Bickmore. BEAT: the Behavior Expression Animation Toolkit. In *Proceedings of SIGGRAPH*, 2001.

[2] J. Cassell, T. Bickmore, M. Billinghurst, 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.

[3] S. Kopp and I. Wachsmuth. Synthesizing multimodal utterances for conversational agents. *Computer Animation and Virtual Worlds*, 15(1):39–52, 2004.

[4] 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.

[5] A. Scheflen. *Communicational structure*. Bloomington: Indiana University Press, 1973.

[6] J. J. Gumperz and J.-P. Blom. Social meaning in linguistic structure: Code-switching

in norway. In J. J. Gumperz and D. H. Hymes, editors, *Directions in Sociolinguistics*, pages 407–434. New York: Holt, 1972.

[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] L. Kovar, M. Gleicher, and F. Pighin. Motion graphs. In *Proc. SIGGRAPH 2002*, 2002.

[9] Y. Li, T. Wang, and H. Y. Shum. Motion texture: A two-level statistical model for character motion synthesis. In *Proc. SIGGRAPH 2002*, 2002.

[10] 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.

[11] K. Pullen and C. Bregler. Motion capture assisted animation: Texturing and synthesis. In *Proc. SIGGRAPH 2002*, 2002.

[12] 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.

[13] Okan Arikan, David A. Forsyth, and James F. O'Brien. Motion synthesis from annotations. *ACM Transactions on Graphics*, 22(3):392–401, 2003.

[14] O. Arikan and D. Forsyth. Interactive motion generation from examples. In *Proceedings of ACM SIGGRAPH 2002*, 2002.

[15] Ken Perlin. Real time responsive animation with personality. *IEEE Transactions on Visualization and Computer Graphics*, 1(1), 1995.

[16] Ken Perlin. An image synthesizer. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, pages 287–296. ACM Press, 1985.

[17] A. Egges, T. Molet, and N. Magnenat-Thalmann. Personalised real-time idle motion synthesis. In *Proceedings of the 12th Pacific Graphics Conference*, pages 121–130, October 2004.

[18] Nick Bobick. Rotating objects using quaternions. *Game Developer*, February 1998.

[19] F. Sebastian Grassia. Practical parameterization of rotations using the exponential map. *Journal of Graphics Tools*, 3(3):29–48, 1998.