A Reusable Model for Emotional Biped Walk-Cycle Animation with Implicit Retargeting

Marco Romeo, Marcelo Dematei, Julian Bonequi, Alun Evans, and Josep Blat

Grup de Tecnologies Interactives, Universitat Pompeu Fabra
Carrer Tanger 122-140, 08005 Barcelona, Spain
`{marco.romeo,alun.evans,josep.blat}@upf.edu`

Abstract. This paper presents a reusable model for rapid animation of the walk-cycle of virtual biped characters, with implicit retargeting of motion capture to any character, regardless of dimensions. Despite modern software continuously improving the quality of automatic assistance, the process of animating a biped character still requires substantial manual intervention. Our research contributes to this field by creating a theoretical model for emotional character walking, defining a series of proportional variables which can be changed to create different emotional walk cycles. We used motion capture data to assign real-world values to these variables, which are then used to procedurally create 'emotional' walk cycles. Due to the fact that we avoid fixed values and work solely with proportions, the system implicitly retargets the data to any biped body shape, regardless of the size and structure of the skeleton.

Keywords: Animation, retargeting, virtual characters.

1 Introduction

Since the idea of technological convergence arose in the early 1990s, the media industry has consistently looked at systems that share resources and interact with each other, cooperating in order to create content in a more efficient and cost-effective manner. This search for more efficient systems is caused, in part, by the nature of digital media production, which remains a very labour intensive, high-risk and high-cost industry. One of the reasons for this is that productions are crafted, almost without exception, at very low levels, in order to better satisfy artistic needs. Indeed, in many applications, the existence of more sophisticated digital tools has actually pushed up costs, as more time is spent on complex off-line processes in the quest for quality.

An excellent example of this can be seen in the design and animation of virtual characters or avatars. These are used in many fields of the audiovisual industry, such as television, video games, internet, and mobile phones. Believable animation (both for bodies and facial expressions) is crucial so that such characters can properly express emotions, improving their ability to communicate. While this issue has been solved in the film industry and, increasingly, in the high-end video games industry, the same techniques cannot be applied to lower-budget productions, due to strong time and hardware resources constraints. Typically, time consumption could be due to

F.J. Perales and R.B. Fisher (Eds.): AMDO 2010, LNCS 6169, pp. 270–279, 2010.

the difficulties in creating a good animation rig for preparing handcrafted deformations that express a character's current emotional state. Those difficulties directly affect the hardware resources as complex animation rigs or a high number of different deformations make real-time animations hard to be replayed while maintaining the desired frame rate. In that sense there is the need for an animation methodology that can rapidly (and in real-time) animate a character by using fewer hardware resources and being sufficiently straightforward for animators to setup.

This paper presents one aspect of our research towards this goal, focusing particularly on the typical human walk-cycle animation. Our approach has been to view the problem from the traditional animator's point of view, attempting to apply well-established philosophies of hand-drawn animation within a modern computing framework. Thus, by studying the relationship between a character's apparent centre of gravity and the curvature of it's spine, we present a new model for walk-cycle animation which classifies the possible walk-cycles into one of four separate gaits.

By studying this model, we define a series of proportional variables of body movement (e.g length of stride in proportion to length of leg). Motion capture data was used to provide real values for the defined variables. As the all the data is stored as proportions, this leads to the creation of an algorithm that allows the implicit retargeting of 'emotional' walk-cycle data to any biped with the correct skeletal structure. The result is real-time modification of a basic walk-cycle animation to allow the character to express a wide variety of emotions while walking. Crucially, our system modifies the walk-cycle while *maintains the underlying animation*, thus if a character is limping, the limp is maintained even though the system may tweak aspects of the animation to change the characters expressed emotion.

We demonstrate the system working both as a plugin for Autodesk 3DS Max (for use by animators) and as a self-contained C++ API, for use in custom real-time graphics engines.

2 Related Work

Early efforts to control the animation of walk cycles were made by both Badler et al.[1] and Hodgins et al.[2], who use similar parameter and goal based techniques that are solved by the animation system. This idea of scripting character animation was extended by Perlin and Goldberg[3], who parameterized human 'actions' and blend the motion data to achieve combinations of movement based on different classes (gestures, stances etc.). A logic system prevents clashes between classes such that a character will not sit and stand at the same time.

The major achievement of these studies was to introduce the idea of abstract descriptions of motions, yet they made little effort to generate those motions (either via motion data or manually created animation). This problem was tackled from a more mathematical point of view by several studies[4][5] that attempted to generate movement based on interpreting the movement curves of the joints of the body. These concepts were further extended by the use of multiresolution filters to modify the 'signals' of the movement curves[6].

Grunvogel et al.[7] introduced the concept of *Dynamic Motion Models,* where the combination of abstraction of movement and procedural generation allows complex animations sequences to be rapidly created. A more general version of this is used by

Abadia et al. [8] who use a dynamic timeline to cue animation clips, and provide the framework for automatic creation of such clips based on an overall emotional theme.

The issue of retargeting of motion captured data was tackled by Meredith and Maddock[9], who use weighted inverse kinematics to adapt motion capture data. This results in individual walk-cycle animation for each character, and enables rapid application of unique characteristics, such as a limp.

Research into procedural walk-cycle animation has differed notably from equivalent work into facial animation, in the sense that it has focused less on the use of emotions. In the facial animation field, perhaps the most cited work is that of Ekman[10], who specified the six basic emotions that can be deduced from facial expressions, independent from cultural background. Densley[11] et al did introduce emotion to the characters appearance, by constraining joint angles based on the emotional state of the character.

In a sense, of all the previous research in this field, it is the work of both Densley et al[11] and Meredith and Maddock[9] that bears the most similarity to the work in this paper. In a sense, our approach combines their philosophies but underpins it with a more formal and focused theoretical background.

As mentioned above, our approach has been to tackle this issue using traditional character animation techniques, applied within a modern computing and mathematical framework. For an introduction to animation techniques and their evolution over the years, both Williams[12], and Johnston and Thomas[13] present the subject in detail.

3 Overview of Approach

Figure 1 is intended as an overview of our approach. A Conceptual Model (introduced in Section 4), based on traditional theory of animation, was used to guide and direct motion capture sessions involving male and female actors (see Section 5). The motion data from these sessions was processed (as described in Section 6) and used as an input for our novel retargeting algorithm. We have created two implementations of this algorithm, one as a plugin for the popular modeling and animation software, Autodesk 3D Studio Max, and the other as separate C++ API than can be used to retarget animation in custom 3D animation or games engine.

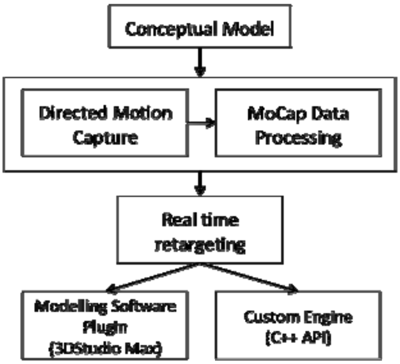

Fig. 1. Diagrammatical overview of the system architecture

4 Conceptual Model

In this section we present our conceptual model that describes how a character's walk cycle may express the character's emotion. Extending some of the principals of the basic walk cycle animation Williams[12], our model is focused on modeling how a character's general posture changes according to the current emotional state. The typical distribution of body mass situates the character's centre of gravity in the lower abdomen. Yet we base our model on the characters *apparent* centre of gravity – the area of the body that, in effect, guides the remainder of the body according to the principals of 'follow-through' animation and 'overlapping action' that are described by Johnston[13]. We use the phrase 'apparent' centre of gravity because we are not physically modeling any changes in the character's actual centre of gravity. In contrast, we are following the traditional animation route of abstracting the concept to a different level in order to allow the animation to better convey the required message.

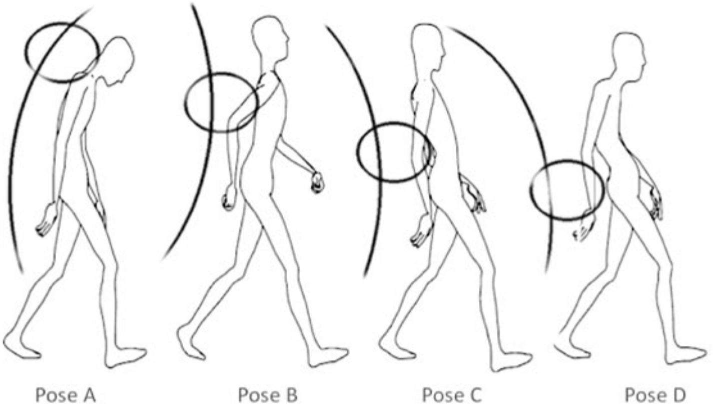

Fig. 2. The four poses of our model. The circle represents the *apparent centre of gravity* – the area of the body that is leading the character while walking. The associated curve shows how the spine is curved in each pose.

Table 1. The four poses of the model, and examples of associated emotions

Pose	Characteristics	Associated emotions
A	Apparent COG is head or neck. Spine curved into a 'C' shape.	Sadness, Depression Concentration/Worry Anger, Hurry Fear
B	Apparent COG at chest height Spine opposite to Pose A	Happiness, Joy, Pride
C	Apparent COG at abdomen Spine curved as B	Satisfaction Relaxation, Serenity
D	Apparent COG dropped to Pelvis Spine curved back further than C#	Sensuality, Arrogance Fear Hurry

Figure 2 shows the four basic poses (or gaits) that we define in our model, based on the location of the characters centre of gravity (COG). Each of these poses associates a particular curve of the spine with a centre of gravity position, which accordingly represents a different manner of walking. The model structures the poses to show the apparent COG dropping in height, from its highest point in the head, to its lowest point in the hips.

So that we could use this model practically in a computer animation system, we translated it into a **series of equations** that describe the movement of the body at each key-frame of the walk cycle[12]. Crucially, all the parameters of these equations are expressed as proportions relating joint position and rotation to the individual dimensions of each skeleton. The list of equations is as follows:

Wrist relative position:

$\mathbf{wrist}.\mathrm{x} = \mathbf{wt}.\mathrm{x} - (\mathbf{ht}.\mathrm{x}+\mathbf{hw}/2)/sw$

$\mathbf{wrist}.\mathrm{y} = (\mathbf{wt}.\mathrm{y}-\mathbf{ht}.\mathrm{y})/al$

$\mathbf{wrist}.\mathrm{z} = (\mathbf{wt}.\mathrm{z}-\mathbf{ht}.\mathrm{z})/ll$

Ankle relative position:

$\mathbf{ankle}.\mathrm{x} = (\mathbf{at}.\mathrm{x}-(\mathbf{ht}.\mathrm{x} + hw/2))/hw$

$\mathbf{ankle}.\mathrm{y} = \mathbf{at}.\mathrm{y}/al$

$\mathbf{ankle}.\mathrm{z} = (\mathbf{at}.\mathrm{z}-\mathbf{ht}.\mathrm{z})/ll$

Hip relative position:

$\mathbf{hip}.\mathrm{x} = \mathbf{ht}.\mathrm{x}/hw$

$\mathbf{hip}.\mathrm{y} = \mathbf{ht}.\mathrm{y}/ll$

$\mathbf{hip}.\mathrm{z} = \mathbf{ht}.\mathrm{z}$

where **wrist, ankle** and **hip** are location vectors;
wt = wrist translation vector; **at** = ankle translation vector; **ht** = hip translation vector;
sw = Shoulder width (Euclidean distance between left and right shoulder);
hw = Hip width (Euclidean distance between left and right joints);
ll = Leg length (Euclidean distance between leg hip joint and ankle);
al = Arm length (Euclidean distance between the shoulder and the wrist).

For joint **rotations**, wrist, spine, pelvis and neck are taken into account. Neck and spine rotations are considered as a concatenation of local rotations.

These equations then form the core of the model, as by assigning different values to the parameters of the equations we can procedurally generate a variety of different animated walk-cycles.

5 Focused Motion Capture and Extraction of Data

While it is possible to fill the equations of Section 4 with randomly selected values, in order to generate meaningful procedural animation, it is necessary to base the values on some form of real data. To achieve this, we carried out a series of motion capture sessions, where male and female professional actors were carefully directed to walk in a manner that expressed the emotions identified from the different poses of the model (see Table 1). However, the motion capture director was careful not to actually instruct the actors the change their performance based on the model. At the time of recording the motion data, the actors were completely unaware of the model, and were merely told to attempt to express the requested emotion through their gait alone. Figure 3 shows example of the emotions recorded during the capture sessions.

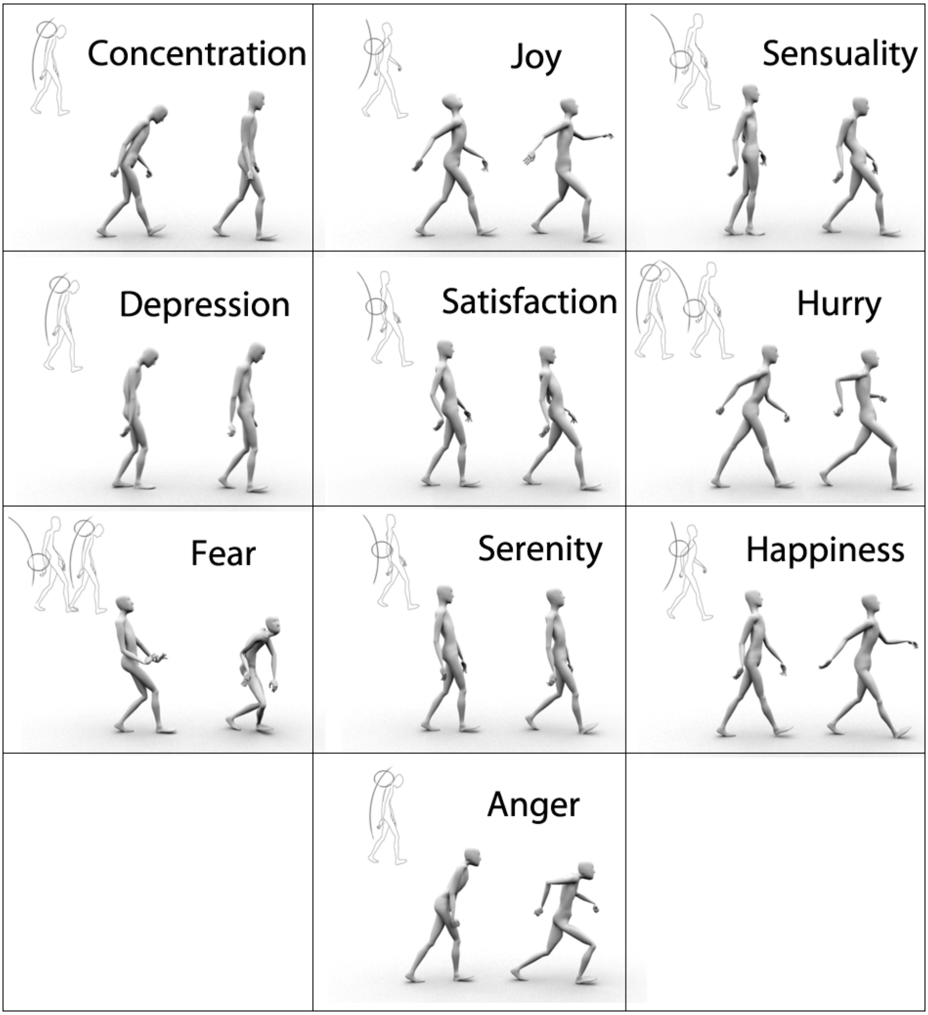

Fig. 3. Still images of real motion capture data, grouped by emotion. Of the two shaded characters in each cell of the table, the one on the right is representing data from the female actor, the character on the left is the male actor. In the upper left corner of each cell is the equivalent pose from the conceptual model (see Figure 2).

6 Implicit Retargeting of Data

Following the motion capture sessions, the recorded data was analysed and the values for the proportional variables for the different emotions were extracted for the five key-frame poses of the typical walk-cycle animation, identified by Williams[12] ("Contact", "Down", "Pass", "Up", "Second Contact"), using the set of equations presented in Section 4.

This resulted in a matrix of proportional values that enable us to mathematically describe the emotions expressed by the actor during the motion capture i.e. a set of values for each emotion. By blending these proportional values into an existing virtual character walk-animation, we are able to transfer the aspects that make the motion captured data "emotional" to the virtual character. It is possible to proportionally blend the data, such that a character can be made to be "a little bit" sad, by blending in only a small percentage of the *Sad* emotion variable set.

Furthermore, there are two very important aspects that separate our system from anything that has been previously produced:

1. **The existing underlying animation of the character is maintained.** The system does not *create* the walk-cycle animation, but modifies the existing animation. This ensures that characters maintain their personality, and any individual quirks added by the animator are not removed.
2. **The dimensions of the character are unimportant.** Because the system is based on the relationship between the proportions of the body, it works on a wide variety of body shapes, so long as the basic skeletal structure is that of a biped.

The equations presented in Section 4 allow values to be *extracted* from the motion capture data. To take those values and apply them to another character (thus carrying out the retargeting) we merely need to inverse the equations to calculate the offsets to the existing animation. As our system only directly controls ankles, hips and wrists, standard inverse kinematics are used to ensure the remainder of the bones of the skeleton move in a believable manner. The side-advantage of this (other than convenience) is that retargeting is *independent of joint-chain length*.

Finally, the system implements time stretching to ensure that the resulting animation matches the average speed of the target emotions (i.e. a sad character should walk slower than a happy one).

6.1 3D Studio Max Plugin

We have created two software implementations for the system, a plugin for 3D Studio Max, and an independent C+ API. Screenshots of our plugin for 3D Studio Max are shown in Figure 4. The plugin allows an animator to easily incorporate the system into an existing walk cycle, using a slider-based GUI to proportionally add or remove aspects of each of the emotions. 10 emotions are explicitly mapped to sliders (those in Figure 3, derived from studying the conceptual model and from studying the literature[10][11][12][13]), by combining the sliders in different proportions, the animator can experiment until the walk-cycle is modified to their satisfaction.

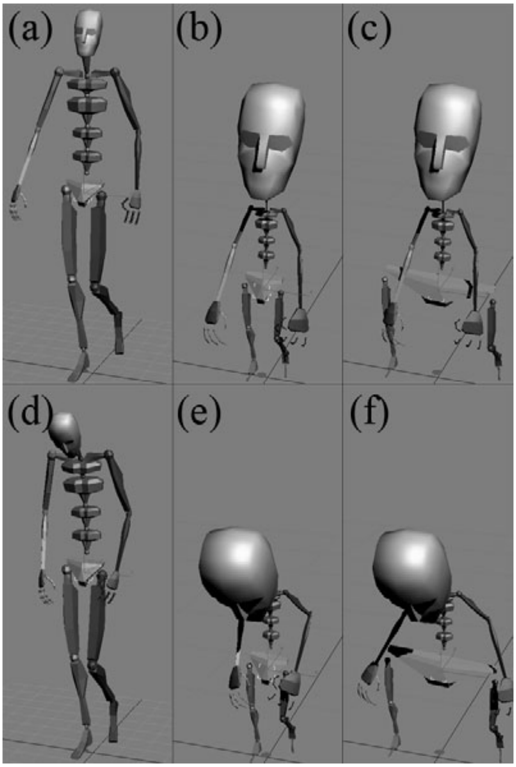

Fig. 4. Screenshots of the results of the plugin developed for Autodesk 3D Studio Max. (a), (b) and (b) show a standard walk animation applied to characters of different dimensions. (d), (e) and (f) show the same characters with the 'Sad' emotion blended in. In (c) and (f), the characters hip joint has been scaled artificially to three times the width, yet the retargeting still works perfectly.

6.2 Independent API

We have also extended the plugin into an independent C++ API that can be used to modify walk-cycles in real time. The API is designed to be used to be used with a variety of graphics and game engines, and we have tested it successfully with two such engines[8][14]. The API gives the developer the option to build in real-time changes to a character's walk-cycle, thus allowing the character to respond directly to actions occurring within the specific application (be they from user input, or from internal factors that may affect the character).

Both the 3D Studio Max plugin and the C++ API are currently being used by our industrial partners in the creation of real-time animated productions.

7 Discussion and Future Work

The contributions of this paper are twofold. First, we present a conceptual model that has enabled us to define a series of proportional variables that describe the biped walk-cycle. Motion capture data was used to creating a matrix of these values for a series of different emotions.

Second, this matrix enabled us to programmatically retarget the emotion-based capture data to any suitable biped walk-cycle, irrespective of character size or shape, and maintaining the underlying animation.

This approach of this system is different to those taken by previous authors. We are not attempting to adapt motion capture mathematically, such as Meredith and Maddick[9]. Neither are we attempting to create a global dynamic motion model such as Grünvogel et al.[7] Rather, we have approached the problem from the practical position of the traditional animator/artist, created a conceptual model, and then used applied modern mathematical and programming solutions to create the final system. This said, the manner in which it has been developed means that it could be easily combined with the weighted inverse kinematic approach of Meredith and Maddock, given that both approaches produce results that *extend* (or tweak) the existing animation, rather than completely replace it.

There are two typical applications of the work, reflected in the two implementations that we have developed. The first is in computer animation for film or television, where our system allows the animator to rapidly add, remove or combine specific emotions to a character, while maintaining the personality that they have created for the that character.

The second, and possibly more powerful, application is for real time situations such as in video games, or 'serious' games for developed for training or educational purposes. Currently in such applications, character emotional state is frequently heavily pre-scripted. Our system opens the door for dynamic linking of emotional walk-cycle animation to non-scripted factors that may affect the 'emotional state' of the character. This enables the character to behave in a more believable way, and reduces the amount of scripted animation work required for the application.

Our future work lies in integrating the system into with other methods in which a character can express emotion, such as static posture, gestures, and facial animation. While systems for both static postures and gestures can be generated using a very similar system that has been presented in this paper, facial animation requires a different approach, and this is the focus of current research.

Acknowledgements

These authors would like to acknowledge the support of the i3Media project and Activa Multimedia Digital, Barcelona, Spain.

References

1. Badler, N., Phillips, C., Lynn, B.: Simulating Humans: Computer Graphics and control. Oxford University Press, Oxford (1993)
2. Hodgins, J., Wooten, W., Brogan, D., O'Brian, J.: Animating human Athletics. Computer Graphics 29, 71–78 (1995)
3. Perlin, K., Goldberg, G.: Improv: A system for scripting interactive actors in virtual worlds. Computer Graphics 30, 205–218 (1996)

4. Witkin, A., Popovic, Z.: Motion Warping. Computer Graphics 29, 105–108 (1995)
5. Unama, M., Takeuchi: Fourier Principals for Emotion-based Human Figure Animation. In: International Conference on Computer Graphics and Interactive Techniques, pp. 91–96. ACM Press, New York (1995)
6. Bruderlin, A., Williams, L.: Motion Signal Processing. Computer Graphics 29, 97–104 (1995)
7. Grünvogel, S., Lange, T., Piesk, J.: Dynamic Motion Models. In: Eurographics (2002)
8. Abadia, J., Evans, A., Ganzales, E., Gonzales, S., Soto, D., Fort, S., Romeo, M., Blat, J.: Assisted Animated Production Creation and Programme Generation. In: ACM Advances in Computer Entertainment Technology, pp. 207–214. ACM Press, New York (2009)
9. Meredith, M., Maddock, S.: Adapting Motion Capture Data Using Weighted Real-Time Inverse Kinematics. ACM Computer in Entertainment 3(1) (2005)
10. Ekman, P., Friesen, W., Ellsworth, P.: What emotion categories or dimensions can observers judge from facial behaviour? In: Ekman, P. (ed.) Emotion in the Human Face (1982)
11. Densley, D., Willis, P.: Emotional Posturing: A Method Towards Achieving Emotional Figure Animation. In: IEEE Computer Animation 1997, pp. 8–14. IEEE Computer Society, Washington (1997)
12. Williams, R.: The Animator's Survival Toolkit. Faber and Faber, London (2001)
13. Johnston, O., Thomas, F.: The Illusion of Life: Disney Animation. Disney Editions (1995)
14. OGRE – Open Source 3D Graphics Engine, `http://www.ogre3d.org`