Expressive Body Animation Pipeline for Virtual Agent

Jing Huang and Catherine Pelachaud

Telecom ParisTech - CNRS, Paris, France
{jing.huang,catherine.pelachaud}@telecom-paristech.fr

Abstract. In this paper, we present our expressive body-gestures animation synthesis model for our Embodied Conversational Agent(ECA) technology. Our implementation builds upon a full body reach model using a hybrid kinematics solution. We describe the full pipeline of our model that starts from a symbolic description of behaviors, to the construction of a set of keyframes till the generation of the whole animation enhanced with expressive qualities. Our approach offers convincing visual quality results obtained with high real-time performance.

1 Introduction

Embodied Conversational Agents are virtual human agents that can communicate through voice, facial expressions, emotional gestures, body movements etc. They use their verbal and nonverbal behaviors to convey their intentions and emotional states. It is necessary for the ECAs to display a large variety of behaviors.

Generating efficient and realistic animations of virtual creatures has always been an open challenge in the computer animation field. Kinematics is a general method for manipulating interactively articulated figures and generating postures. In computer graphics, articulated skeleton models are used to control virtual vertebral living creatures, such as human beings or animals, which appear frequently in films and video games. Inverse Kinematics (IK) is a method for computing joint rotation values of individual degree of freedom via predefined rotation and position constraints. In most of existing systems, animation of body parts is done independently of each other. For example, an arm gesture and torso movement are computed separately and then combined. Such computation gives rise to unnatural animation and stiff-looking creatures. ECAs are interactive entities; all their animations have to be rendered online.

Our work focuses on the realization of animation for virtual conversational agents. Our animation model takes as input a sequence of multimodal behaviors to generate. It relies on a hybrid kinematics solution for generating full body posture. Moreover our solution generates two types of motion. On the one hand it computes all movements specified over each modality (eg arm, torso or head movement). On the other hand, it also considers movements arising from other movements. For example it will generate a torso and shoulder movement resulting

Y. Nakano et al. (Eds.): IVA 2012, LNAI 7502, pp. 355–362, 2012.

from a given arm movement (eg, when the arm has to reach a distant point in space). Considering independent and dependent movements gives rise to a more realistic and natural animation. Our model embeds also an expressive module with qualitative variations of body movements. Our algorithm is efficient; it can generate realistic whole body animations in real-time.

In the remainder of the paper, we first present existing approaches, then turn our attention to our approach. In section 3, we describe our animation pipeline with our expressivity model. We illustrate our explanation with examples.

2 Previous Work

In this section we will present different works regarding expressivity and animation computations that are related to our work. Several approaches have been proposed to model expressive behaviors. The EMOTE system [3] introduced low level parametrization to generate expressive gesture. The parameters are abstracted from Laban principles (1988). The Greta system [14] [4], defines a low level parametrization derived from psychology literatures. In the realization of animation, both of these works decomposed character skeleton into small parts (the head, the torso, the arms, etc) and solved the system by different controllers acting locally. Such an approach does not allow modeling motion propagations, ie how motion over one modality may affect another one. In our work, we choose to deal with whole body motion with a global view.

Michael Neff *et al.* [11] [12] [13] presented their aesthetic motion generation system. Their model starts from a high level expressive language that is translated into precise semantic units that can be simulated by physical or kinematics methods. The translation mechanism encloses a selection and a refinement steps for choosing the gesture movements.

SmartBody system [17] proposed a controlling system that employs arbitrary animation algorithms. The system can schedule different task controllers, it allows realizing modular animation control and propagating motions over the body parts. As SmartBody, we achieve a whole body management from a lower level. But our approach is based on the hierarchical dependency of the agent's body structure.

The EMBR system [6] offers an animation pipeline with their motion factory, scheduler and pose blender modules. The animation to be realized is described with the EMBR script. It can deal with different formats of animation (IK, motion data, etc). Our system follows a similar animation pipeline. However our pipeline solves the conflict for body parts. Indeed, our system is based on IK techniques. IK is used to do retargeting, and it makes the final decision for key postures. Several inverse kinematics [1] [2] [5] methods are also proposed to achieve reaching tasks. Such methods need to calculate the whole body posture in realtime for a given trajectory of the wrist position in space.

Tan [16] talks about the importance of using the postural expressions to express action tendencies. Meanwhile, behavioral studies on posture have also been

made [7]. Although their studies are based on static postures, the authors noticed that expressive postures are rather important. They also note that generating automatically variations of expressive postures is useful for simulating human-like animation.

3 Implementation of Animation Pipeline

We have implemented a framework of embodied conversational agents that respects the SAIBA [9] framework illustrated in Figure 1. Our framework takes as input a file described with FML the standard Functional Markup Language which defines the intentions and emotional states in a high level manner. The Behavior planner translates the FML tags into sequences of standard BML, Behavior Markup Language, entries. Sequences of time-marked BML-like signals are instantiated within the Behavior Realizer.

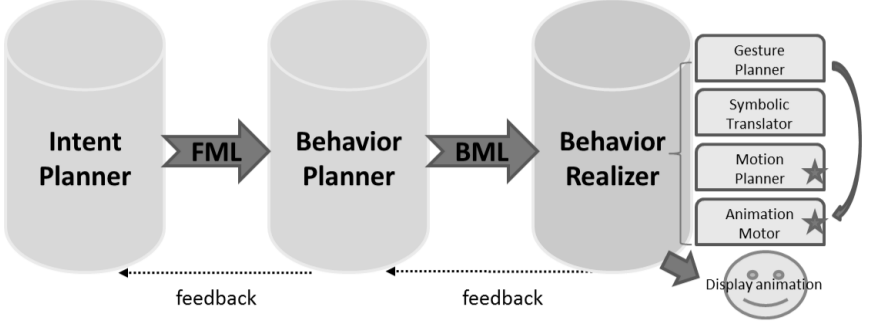

Fig. 1. The standard SAIBA framework defines the modularity, functionality and the protocols for ECAs. In the Behavior Realizer module, the parts with a star correspond to our work in the animation pipeline.

3.1 Overview of Our Pipeline

In this paper, we present the implementation of our animation pipeline. It starts by receiving BML-like symbolic signals time stamped in the motion planner. All signals are received by streaming, and hence our animation computations need to be achieved on the fly. Each signal (hand, torso, head, etc) includes gesture phases, expressivity parameters, gesture trajectory and the description of shape and motion. As shown in Figure 2, signals can be scattered (step 1) from different modalities, ie torso movements, head movements, hand gesture movements (two groups: left and right sides). After receiving scattered signals, we apply our pipeline to generate our animation sequence illustrated in Figure 2.

In the remainder of this section we detail each step of our pipeline. Expressivity parameters are presented in section 3.5.

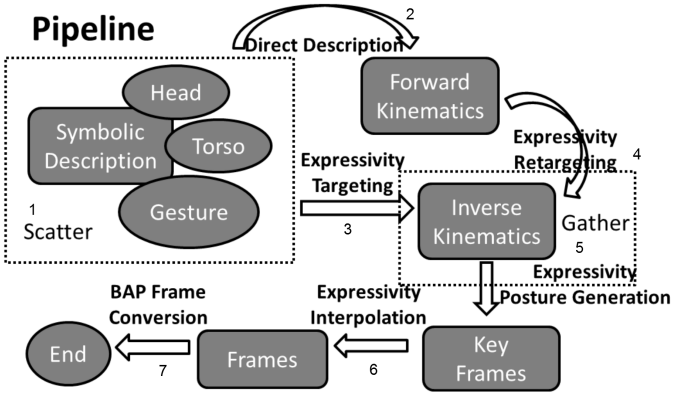

Fig. 2. The animation pipeline takes as input sequences of symbolic gestures. It computes the whole expressive animation; each step is numbered in this figure.

3.2 Targeting Process

The targeting process describes the hand gesture trajectory. This is often referred to the "Path driven" approach. Path form, such as line, circle, can be defined by mathematical functions. For building these path forms without the dynamic branching, we chose to use an approximation of a sinus function: $f(t) = R * sin(T * \pi * t + Shift)$, where R is the amplitude that defines the radius of local circle, T is the temporal variation of frequency, $Shift$ controls the path direction. The final path position P is defined as $P = P_{center} + P(f(t_x), f(t_y), f(t_z))$. By varying the 3 parameters, we can construct different gesture paths. For example, for linear path, T value can be just set to zero. Some other possible gesture paths are saved as sequences of key points corresponding to 3D positions in files.

3.3 Gathering Process

The purpose of the gathering process is to generate the full body key frames that corresponds to body postures. We chose to compute the full body posture as a whole and not as a concatenation of body parts positions as it allows capturing dependent movements across body parts. For example, reaching hand or gaze targets may affect torso movements. When looking on the far left, head, shoulder and torso are all turned to the left.

We sort out sequences of all body parts into one list of key frames ordered by time markers. Information of each key frame is filled with the information from the body part sequences. To complete the key frame specification, we can use either the "lazy" approach or the "complex" approach that are illustrated in Figure 3. The "complex" approach considers that all the movements are equally important, then we need to fill all the body parts for each key frame by interpolating between the previous element and following element of its sequence (linear interpolation); On the other hand, the "lazy" approach privileges some movements. E.g., if a key information has only torso movement, the torso movement

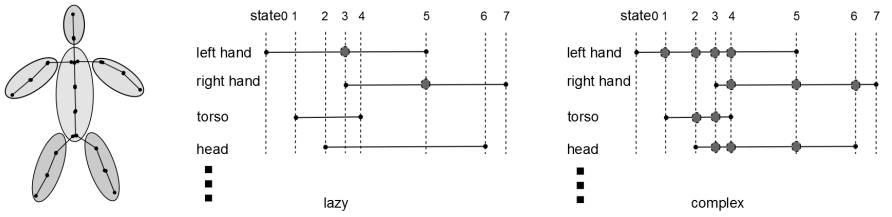

Fig. 3. (Left): shows the character skeleton decomposition, **(Middle):** shows the lazy approach. **(Right):** shows the complex approach.

is dominant, we can only apply torso movement; if a hand gesture is missing, but for one given sequence animation, it is very important, then we must fill it. For the lazy approach, we only fill in existing parts for each key frame and interpolate the important missing movements. The importance is defined by priority parameters. The lazy approach is flexible and has less computation. The resulting key frame list is used in the posture generation stage for creating posture key frames.

3.4 Posture Generation

To compute a body posture, we can apply forward kinematics, inverse kinematics or a mix of both techniques. The forward kinematics uses the description defined in the gesture phases to realize the initial states of the key frames.

Then, when needed, we retarget the gestures by using inverse kinematics. A hybrid inverse kinematics solution is used to solve dependency between body movements. For head and torso movements, we use simple analytical methods. For shoulders and hand gestures, we use our constrained mass spring IK solver. However we have to make sure that all targets are in the reaching space of the arms combined with the torso. If the target is too far away, the movement is transformed as hands pointing in the direction of the target. We do not consider foot step forward to solve this situation.

More concretely, we start by computing the potential target of torso. We check the targets of both hands $T1, T2$. We look if the hands movements needs torso movements. The arm length l_{arm}, the vertebrae (vt1, vl5 in MPEG4 H-ANIM) positions P_{vt1}, P_{vl5} are already defined in the skeleton system. The horizontal pointer of torso is the direction of the center of hands reaching targets. The amount of vertical lean is computed by using trigonometry approximations. Then we compute the hand gestures with this new torso posture. We apply our IK solver on the arm chain, starting from the sternoclavicular till the wrist. After this IK stage, we generate the animation sequence of the key frames using simple joint rotation informations. We use the quaternion based spherical linear interpolation(slerp) to generate all frames, and convert them into "BAP" frames (Body Animation Parameter (MPEG-4) frames).

3.5 Expressivity

We have defined a set of expressivity parameters to modulate the quality of body movements. We group these parameters into 3 sets: Gesture Volume controls the spatial variation of gestures; Sequential Variation controls the time based variation; Power variation allows us to further control the dynamism of these movements.

Gesture Volume: The idea of Gesture Volume is to use certain parameters to control the spatial form of posture shapes. The Spatial parameter is used to control the variation of McNeill's sectors [10]. We have two levels of control of these sectors. The first one is based on the spatial parameter which will scale the sectors using the same method as [4]. The second level adjustment depends on the torso position. The sector centers would be influenced by a scale factor and a rotation factor which are given by the torso. The scale factor $s = (P_{vt1'} - P_{vl5'})/(P_{vt1} - P_{vl5})$ is the ratio between the new length and the old length of the torso. The rotation factor is the rotation value applied on vl5 after the forward kinematics stage, which is used to change the sector box's orientation.

We also have the openness parameter that changes the gesture form computed in the IK stage. Its value ranges between 0 and 1. A small value tightens the body; it makes the elbow closer to the center; and a big value increases the space between the arm and the torso as illustrated in Figure 4 on the left. The openness parameter affects the orientation of the elbow swivel angle [18]. A larger openness value also applies a bigger rotation on the torso that indirectly releases the tension of the arms as shown in Figure 4 on the right.

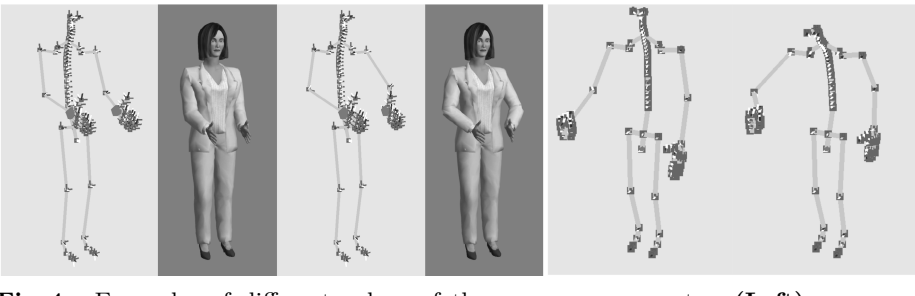

Fig. 4. Examples of different values of the openness parameter: **(Left)**: openness influences the gesture form. **(Right)**: openness influences the torso position.

Sequential Variation: Three parameters, "Fluidity", "Power" and "Tension", are used to modulate the gesture path as well as its timeline. Our signals are already time-stamped in the scatter module (see Figure 2). "Fluidity" refers to the degree of continuity of a movement. To simulate it, we use similar idea as proposed by [3] [4] to parametrize the Kochanek Bartels splines(TCB splines) [8]. We define "Power" as a force that changes implicitly the speed with certain acceleration. "Tension" [15] describes the amount of energy that has to be expended for some positions but not others, and hence more effort needs to be exerted for the gesture to keep its original position. These two last parameters are simulated

by varying the bias parameter and the tension parameter of TCB splines. We also simulate accelerations by using some easing in-out functions to change time stamps for key frames (interpolation for target path) and frames (interpolation for joint rotation).

Additions: Power Variation: In our system, the power parameter does not only affect gesture paths, but also key frame postures. It means that a larger value of power influences more body parts. For example, a movement of the arm done with a high power will affect also shoulder and torso movements. This propagation of movements between body parts is possible as we use a full body IK framework. Torso can be affected by hand gestures as we define target energies. It is a similar idea as the constraints priority [1]. Our hybrid solver builds upon an interactive manner which is controlled by the "Power" parameter. As "Power" increases, it can influence the whole body gesture, both shoulders and torso.

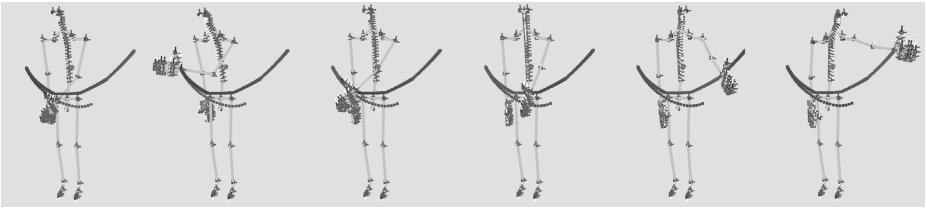

Fig. 5. One sequence animation using IK postures of our skeleton

3.6 Result of Virtual Agent

Our pipeline has been integrated into our virtual agent framework. The skeleton system is based on MPEG-4 H-ANIM 1.1 model. Our virtual agent system takes as input an FML file. It instantiates this input file into a sequence of BML tags. This sequence is the input of our animation pipeline.

Our solution encompasses two passes: forward kinematics specified by the BML tags; inverse kinematics that further influences the body depending on the energy descriptions and the expressivity parameters. We can note that torso and shoulder movements can be automatically generated due to movement propagations. Such movement can happen implicitly without any torso movements defined by BML tags, as illustrated in Figure 5.

4 Conclusion

In this paper, we have presented a full body expressive animation pipeline. We have proposed a hybrid approximation for posture computation based on the dependency of body parts. Our pipeline is compatible with existing target reaching models. Our expressive posture method provides high quality visual results in real-time. This system offers more flexibility to configure expressive FK and IK. It can be extended to other articulated figures.

Acknowledgements. This work has been funded by the French National Research Agency (ANR, for the project: CeCil).

References

1. Baerlocher, P., Boulic, R.: An inverse kinematics architecture enforcing an arbitrary number of strict priority levels. Vis. Comput. 20(6), 402–417 (2004)
2. Boulic, R., Thalmann, D.: Combined direct and inverse kinematic control for articulated figure motion editing. Computer Graphics Forum 11(4), 189–202 (1992)
3. Chi, D., Costa, M., Zhao, L., Badler, N.: The EMOTE model for effort and shape. In: SIGGRAPH 2000, New York, USA, pp. 173–182 (2000)
4. Hartmann, B., Mancini, M., Pelachaud, C.: Implementing expressive gesture synthesis for embodied conversational agents, pp. 188–199 (2006)
5. Hecker, C., Raabe, B., Enslow, R.W., DeWeese, J., Maynard, J., van Prooijen, K.: Real-time motion retargeting to highly varied user-created morphologies. ACM Trans. Graph 27(3), 27:1–27:11 (2008)
6. Heloir, A., Kipp, M.: EMBR – A Realtime Animation Engine for Interactive Embodied Agents. In: Ruttkay, Z., Kipp, M., Nijholt, A., Vilhjálmsson, H.H. (eds.) IVA 2009. LNCS, vol. 5773, pp. 393–404. Springer, Heidelberg (2009)
7. Kleinsmith, A., Bianchi-Berthouze, N.: Recognizing Affective Dimensions from Body Posture. In: Paiva, A.C.R., Prada, R., Picard, R.W. (eds.) ACII 2007. LNCS, vol. 4738, pp. 48–58. Springer, Heidelberg (2007)
8. Kochanek, D.H.U., Bartels, R.H.: Interpolating splines with local tension, continuity, and bias control. In: SIGGRAPH (January 1984)
9. Kopp, S., Krenn, B., Marsella, S.C., Marshall, A.N., Pelachaud, C., Pirker, H., Thórisson, K.R., Vilhjálmsson, H.H.: Towards a Common Framework for Multimodal Generation: The Behavior Markup Language. In: Gratch, J., Young, M., Aylett, R.S., Ballin, D., Olivier, P. (eds.) IVA 2006. LNCS (LNAI), vol. 4133, pp. 205–217. Springer, Heidelberg (2006)
10. McNeill: Hand and Mind: What Gestures Reveal About Thought. The University of Chicago press, Chicago (1992)
11. Neff, M., Fiume, E.: Modeling tension and relaxation for computer animation. In: SCA 2002, pp. 81–88. ACM, New York (2002)
12. Neff, M., Fiume, E.: Artistically based computer generation of expressive motion. In: Proceedings of the AISB, pp. 29–39 (2004)
13. Neff, M., Fiume, E.: AER: aesthetic exploration and refinement for expressive character animation. In: SCA 2005, pp. 161–170. ACM, New York (2005)
14. Niewiadomski, R., Bevacqua, E., Le, Q.A., Pelachaud, C.: Cross-media agent platform, pp. 11–19 (2011)
15. Edwards, A.D.N., Harling, P.A.: Hand tension as a gesture segmentation cue. In: In Proceedings of the Progress in Gestural Interaction, pp. 75–88. MIT mimeo (1997)
16. Tan, N., Clavel, C., Courgeon, M., Martin, J.-C.: Postural expressions of action tendencies. In: Proceedings of the 2nd International Workshop on Social Signal Processing. ACM, New York (2010)
17. Thiebaux, M., Marsella, S., Marshall, A.N., Kallmann, M.: Smartbody: behavior realization for embodied conversational agents. In: Proceedings of the 7th International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS 2008, vol. 1, pp. 151–158 (2008)
18. Tolani, D., Goswami, A., Badler, N.I.: Real-time inverse kinematics techniques for anthropomorphic limbs. Graph. Models Image Process (2000)