Animation Guidelines for Believable Embodied Conversational Agent Gestures

Ivan Gris, Diego A. Rivera, and David Novick(✉)

Department of Computer Science, The University of Texas at El Paso,
500 West University Avenue, El Paso, TX 79968-0518, USA
ivangris4@gmail.com, darivera2@miners.utep.edu,
novick@utep.edu

Abstract. In animating embodied conversational agents (ECAs), run-time blending of animations can provide a large library of movements that increases the appearance of naturalness while decreasing the number of animations to be developed. This approach avoids the need to develop a costly full library of possible animations in advance of use. Our principal scientific contribution is the development of a model for gesture constraints that enables blended animations to represent naturalistic movement. Rather than creating over-detailed, fine-grained procedural animations or hundreds of motion-captured animation files, animators can include sets of their own animations for agents, blend them, and easily reuse animations, while constraining the ECA to use motions that would occur and transition naturally.

Keywords: Embodied conversational agents · Animation · Usability

1 Introduction

Animation of embodied conversational agents (ECAs) too often seem stilted or unnatural, despite animation's history as an art that has been practiced for over a century. In the movies and television, animation grew from a novelty to a major form of artistic expression, with major influence beyond the confines of the animation frame. Actors use principles and techniques from animation during their performances, and artists borrow elements from those performances to make their characters realistic. Cinematography, through the use of visual tricks, camera movements, actor training, and special effects, has created intricate visual storytelling elements that are usually not present in ECA systems. Perhaps ECAs do not feature realistic animation because there is little need for naturalistic animation in most computer applications. Most commercial computer applications are metaphor-driven, where users interpret an icon's image and associate it with its function without requiring a highly realistic animation. For example, users do not see or need a hyper-realistic 3D model of animated scissors every time they click on the "cut" function in common office applications. Likewise, users do not expect realistic animations for characters, when interpretation alone can suffice. Instead, users have come to expect non-realistic characters with limited functionality for whose actions users have the responsibility to interpret.

© Springer International Publishing Switzerland 2015
R. Shumaker and S. Lackey (Eds.): VAMR 2015, LNCS 9179, pp. 197–205, 2015.
DOI: 10.1007/978-3-319-21067-4_21

Ironically, as agents become more realistic, the unnaturalness of their animations can become more evident. Consequently, for human-ECA interactions to become more natural, agents will have to be more successful in interpreting users' gestures and the agents themselves will have to be more realistic in their selection and performance of gestures.

While some animation systems produce realistic and smooth movement for ECAs, research and development of the agents typically focus on the appropriateness or categorization of gestures in taxonomies that define what should be displayed in response to different user behaviors. The responsibility for animation quality is often delegated to an artist, who is limited by the requirements of the particular agent and who has little creative freedom to improve the expressiveness of gestures. As a practical matter, this approach produces rough-and-ready results but does not provide specific guidelines or replicable techniques for producing ECA animations that are appropriately realistic. What is needed, then, is a systematic approach to realistically natural gesture animations for ECAs that can be applied by developers who are not experts in animation.

To respond to this need, we developed an automated animation system that can create a wide range of realistic animations based on a small set of states that can be blended, combined, layered, and transitioned. Our approach is based on an adaptation of principles of lifelike animation and is implemented via an animation graph in Unity's Mecanim system. Our approach contrasts with that of other ways of specifying ECAs and their interactions. Systems such as BEAT (Cassell, et al. 2004) and SPARK (Vilhjálmsson 2005) provided a remarkable amount of detail for gesture based on discourse analysis, but unfortunately these approaches require large sets of micro-animations. And trying to save effort by generating these animations procedurally causes the animations to appear robotic and unnatural.

We seek to help developers and researchers with little background in 3-D modeling or animation to create more natural movement for their agents without requiring fine-grained motion capture. In this paper, accordingly, we review the principles of animation and adapt them to ECAs, describe the mechanism of our approach to producing natural animations based on these principles, describe the collection of a gesture corpus and the development of animations based on this corpus, and conclude with a brief discussion of future work.

2 Animation Principles

Traditional animation theory has suggested twelve principles for creating "the illusion of life" (Thomas & Johnston, 1981). In order of importance, the twelve principles are squash and stretch, anticipation, staging, pose to pose, follow-through or overlapping action, slow in and slow out, arcs, secondary action, timing, exaggeration, solid action and appeal. Although these guidelines were meant for artists, applying the principles to ECAs highlights flaws in standard approaches to animating agents. Accordingly, we describe in detail each animation principle and how it can be applied to ECAs.

Squash and Stretch. This principle provides the illusion of weight and volume. It is particularly useful in animating dialogue and facial expressions. In artistic terms, this is

the most important factor, as it ensures that the character does not change shape and represents the character's weight. For ECAs, it is often preferable to simulate real-world physics so that users can better relate to the agent by acknowledging that the same real-world rules apply for both human and ECA. This principle might not apply to agents that are not human-like.

Anticipation. This principle involves the character's movement in preparation for major actions that they are about to perform. Common examples include running, jumping, or even a change of expression. Almost all real action has major or minor anticipation, yet this is one of the most overlooked animation principles in ECA development. Agents often need to react in real time to users' actions, often involving both speech and gesture. So by the time a system recognizes the user's communicative behaviors and formulates an appropriate response, the system is likely to have to perform immediately the agent's main gesture motion response, leaving no time to perform the anticipation animation. This is a key cause of agent actions appearing to be robotic, as it creates a move instantaneously with seemingly no previous thought or intent. To overcome this obstacle the system has to have a set of anticipation animations that can be used flexibly by predicting the animation that will be required; even as a broad guess can provide a more realistic animation through anticipation than omitting the anticipation animation.

Staging. This principle states that a pose or action should clearly communicate to the audience the mood, reaction, or idea of the character as it relates to the story and its continuity. Staging, in other words, directs the audience's attention to the story or the idea being told. This represents a problem for ECAs because cinematography often uses camera angles and close-ups to make certain story elements salient for the audience. In addition, care must be taken when building scenery and backgrounds so that they do not compete with the animation for the audience's attention. The main problem with staging, though, is that ECAs often do not have a proper stage on which to perform. This is a design issue that should be addressed early in development. In experiments (Gris et al. 2014), we have used staging techniques by providing our agents with a virtual environment. This approach has led to decreased memory loads and higher attention rates (Gris et al. 2014).

Straight Ahead and Pose-to-Pose Animation. Due to the nature of 3-D animation software, we use pose-to-pose animation via key frames. But this does not mean that that animations should be planned simply by creating an initial pose and a final pose, and then letting the 3D software interpolate the sequence automatically. Planning of animations should include transition between poses in the most natural way possible. To achieve this seamless effect, animations should be designed with the proper length, looping poses, and interruption segments so that animations can be combined or transitioned at any time.

Follow-Through and Overlapping Action. In simple terms, this principle means that nothing stops all at once. When an agent is moving and the animation ends, the agent cannot simply go to a static pose. Rather, it should blend with a weight shift or another appropriate motion that provides the illusion of continuous realistic movement, even though the main underlying action has stopped. Following this principle can eliminate unnatural movements such stopping in the middle of a step while walking.

Slow in and Slow Out. This is one of the most important principles for agents who do conversational turn-taking. This principle enables animators to soften the animation and make it more lifelike. Similar to anticipation, attention is drawn at the beginning of the animation and at the end of the animation, as these are often indicators of turn-taking. At the beginning of the motion, people will usually try to guess what the reaction will be, and at the end people need an indication to know that the agent is about to finish and that the users will able to jump into the interaction again.

Arcs. In 3D modeling all organic characters, including humans and animals, are made of curves. In contrast, robots are made of sharp edges. Animating organic characters follow arcs or slightly circular paths because of the nature of our joints. This principle applies to almost all physical movement, including turns, eye motion, wand walking paths. Accordingly, movements performed in a straight line will seem robotic or unnatural.

Secondary Action. The principle of secondary action applies to actions that enrich the main action. Secondary action adds character. For example, envision a male character about to invite a female character on a date. If it approaches the female character slowly often changing direction, it gives the impression of a shy and unsure character. However envision that same character with many concurrent secondary actions, such as fidgeting and frequent gaze movement away from the target character. These actions may not be necessary, but they enhance the animation by making it more obvious, clear, and natural (Lasseter 1987).

Timing. This is a highly subjective principle for animations that establish mood, emotion, and character reaction to the situation. For example, a pointing animation displayed in a faster timing can indicate a sense of urgency, while a slow pointing animation can indicate laziness or lack of interest. Most animation systems for ECAs permit developers to specify the time the animation should be running, but it is harder to find systems that can accelerate the animation, which is equally important.

Exaggeration. In a play, actors do not behave like normal human beings. Instead, they exaggerate their gestures to make them observable by the audience. While developers of agents commonly to try to make the animations as humanlike as possible, there is a potential benefit in exaggerating animations to make them more noticeable. Of course, one must be careful not to over-exaggerate; the risk is that of becoming too theatrical or excessively animated. Although some research has examined how big animations should be, based on the introversion or extraversion of the subjects (Neff et al. 2010; Hostetter et al. 2012), the field still lacks detailed guidelines for gesture amplitude. Artists' suggestions to use good taste and common sense are not very helpful in this regard. Moreover, perceived gesture amplitude depends on the distance from the display, the type of media, and the screen size (Detenber et al. 1996; Loomis et al. 2003).

Solid Drawing. In the 1930 s, animators used this approach to create a three-dimensional space from the two dimensions of drawings on paper. In the case of animated agents, though, this principle is redundant because agents' 3-D models already include solid drawing.

Appeal. This principle usually refers to visual attractiveness, but it can also be used as part of the character design to provide an agent with clear intentions. This is another highly subjective trait that should not be underestimated. In one of our pilot studies, users perceived an agent who was supposed to help users in a game scenario as being actually a potential enemy. Our intention was, of course, to convey a helpful, friendly character. But due to our lack of attention to the agent's visual appearance, our animated agent conveyed the opposite impression. In general, this is a trait that combines visual representation, dialog, context, and gestures, and it is difficult to achieve when any of these elements is missing.

Although all twelve principles were developed originally for traditional animation and to design characters for passive audiences that do not interact with agents, they can still help in developing agents that are more relatable, believable, and accurately representing what we want our agents to convey.

3 Animation Blend Tree

To enable developers of ECAs to use the twelve principles of animation, as adapted for agents as described in Sect. 2, we developed an animation design approach implemented as a tree of blendable animations. Our system, based on Unity's Mecanim, creates an autonomous animation workflow. We describe the principles of our animation tree and how its animations can be blended to create a very large number of naturalistic movements.

In Mecanim, the animation tree is a state machine where each state contains an animation and a weight. The weights of the animations are used to transition between them or blend animations that affect the same body parts. When there is a single animation being executed, its weight is equal to one. Every animation that is not being executed has a weight of zero. Each state transition transfers the weight of the current state towards the next intended state, until the new state has a weight of one and the previous state has a weight of zero. This means that at the midpoint of the transition each animation is being displayed with half the strength, containing properties of both animation states simultaneously. These transitions are not themselves states but rather the equivalent of a cross-fade between animations.

Using Mecanim's animation tree, we created, analyzed, and weighted a set of animations so that each state transition enforces adherence to the animation guidelines. That is, the structure and design of our implementation of the animation tree enables end users to link animations together while limiting the linking of unnatural animations (e.g., going from the running state to the crouching state), taking care to make transitions realistically gradual. Each state can include movements from any or all of the body-part layers listed at the top left of the figure. The blend tree can be expanded to include more movements.

In addition, our tree contains a set of layers that provide additional control for running animations procedurally in real time by enabling users to specify animations for fingers and hands, facial expressions, and upper and lower body separately. This effectively enables us to combine a happy facial expression while pointing with the left hand and walking, where otherwise it would require a specially designed animation that

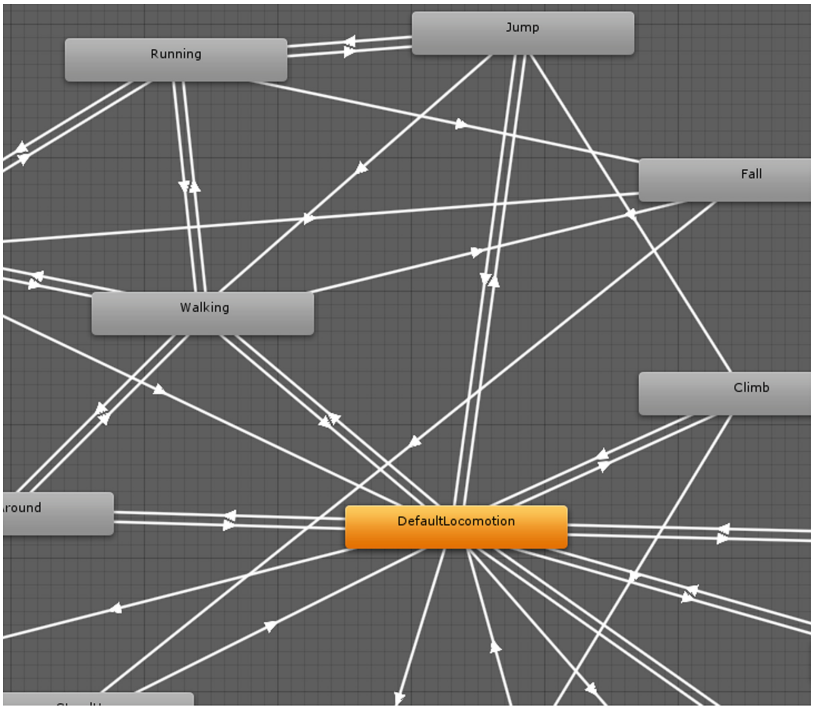

Fig. 1. A section of the animation tree detailing the permitted locomotion transitions for lower body movement. The labels of the animations differ in style (e.g., "Running" vs. "Jump") because some animations are continuous loops (e.g., running, walking), and others are one-time movements (e.g., jump, stumble).

would be useful only in this particular situation. Although it can be argued that crafting specific animations for each situation produces a more accurate representation of the intended movement, creating all of an agent's possible animations in this way be impossible because the combinatorial explosion of the movements of the different parts of the body. In our approach, the goal of the layers is to enable a maximum number of combinations from a small subset of independent general animations, while maintaining the naturalness and movement flow described in the twelve guidelines.

Figure 1 presents a subset of our animation tree. (We note that animation tree is the official name for Unity Mecanim structures, but our modifications effectively turn this "tree" into a graph.)

4 Methodology and Corpus

Given Mecanim's basic architecture, the twelve animation principles, and our proposed solution to the combinatorial explosion, we developed our particular animation tree for ECAs based on a series of observations of natural conversations. Our corpus contained twelve dyadic conversations. Each conversant was recorded in an otherwise empty

Fig. 2. From left to right, video and depth noise from the participant, motion capture data, and agent with animation data incorporated. The top right displays a thumbnail view of the raw front and side video as processed with iPi Soft.

room using two Kinect sensors that recorded side and front views for each participant, who stood facing each other. Each recording contained video and depth information, and this information was interpreted by motion capture software and translated to one of our agents, a full-body virtual 3D character. Each animated file was then empirically analyzed and categorized based on movement similarity. Figure 2 shows the stages of motion capture.

Because we sought to preserve naturalness of movement, we focused on the mechanics of the gesture rather than the gesture itself. That is, we studied the movements preceding the main gesture. For example, before performing an action such as pointing, leaning, or explaining, participants often prepared for several seconds before performing the intended gesture. Based on these observations, we classified the possible transition points between gestures, eliminating many possible but unnatural combinations that violated the animation principles. We also layered the captured animations by dividing the virtual character's body into lower body (locomotion), upper body, head, face, and hands. Additional layers were created for lip-sync and blinking controls; these, however, did not come from the motion capture analysis. Figure 3 presents the animation layers corresponding to the agent's body regions.

Fig. 3. To the left, the list of layers affecting different body areas. These layers can be blended with any or all others with the exception of the special animation. To the right, the list of parameters that characterizes the current running animation.

The *Special* layer in Fig. 3 is reserved for when there is the need to create an animation that is superimposed over all others—for example, a backflip, dancing, or other activity that requires simultaneous full-body coordination. Parameters detect the last pose into which animations are blended. If an animation has taken place but has not transitioned to a new state, the tree limits available movements. For example, if the agent sat down, it must remain sitting and cannot perform any other lower body animations until it stands up. In our approach, exceptions to this rule—allowing an agent to sit while pointing or explaining, for example—must be explicitly identified via connections in the tree.

5 Conclusion

While our animation tree is based on subjective principles, these principles reflect characteristics of movement that are consistently present in everyday interaction. By examining motion capture data from people engaged in dyadic conversation, we can observe, classify, layer, and replicate these animations. We can then infer a set of rules, based on the adaptation of the twelve animation principles as they apply to ECAs, and enforce them through our animation controller structure. The animation tree serves as a starting point for more elaborate animation patterns by helping developers with little animation knowledge set up realistic full-body animations, while retaining the potential to expand the solution to fit more unusual or additional cases.

We are currently using this system to create our next generation of ECA applications, which require real-time reactions to speech and gesture input, mimicry of gestures, and natural-looking movements. Future work includes creating a public animation library. We are also working on releasing the core version of our animation tree, which would enable non-animators to develop realistic movement for human-ECA interaction with minimal coding and no knowledge of animation.

References

Bates, J.: The role of emotion in believable agents. Commun. ACM **37**(7), 122–125 (1994)

Cassell, J., Vilhjálmsson, H.H., Bickmore, T.: BEAT: the behavior expression animation toolkit. In: Cassell, J., Högni, H., Bickmore, T. (eds.) Life-Like Characters, pp. 163–185. Springer, Berlin Heidelberg (2004)

Detenber, B.H., Reeves, B.: A bio-informational theory of emotion: Motion and image size effects on viewers. J. Communi. **46**(3), 66–84 (1996)

Gris, I., Novick, D., Camacho, A., Rivera, D.A., Gutierrez, M., Rayon, A.: Recorded speech, virtual environments, and the effectiveness of embodied conversational agents. In: Bickmore, T., Marsella, S., Sidner, C. (eds.) IVA 2014. LNCS, vol. 8637, pp. 182–185. Springer, Heidelberg (2014)

Hostetter, A.B., Potthoff, A.L.: Effects of personality and social situation on representational gesture production. Gesture **12**(1), 62–83 (2012)

Lasseter, J.: Principles of traditional animation applied to 3D computer animation. ACM Siggraph Comput. Graphics **21**(4), 35–44 (1987)

Loomis, J.M., Knapp, J.M.: Visual perception of egocentric distance in real and virtual environments. Virtual Adapt. Environ. **11**, 21–46 (2003)

Neff, M., Wang, Y., Abbott, R., Walker, M.: Evaluating the effect of gesture and language on personality perception in conversational agents. In: Safonova, A. (ed.) IVA 2010. LNCS, vol. 6356, pp. 222–235. Springer, Heidelberg (2010)

Thomas, F., Johnston, O., Frank, T.: The Illusion of life: Disney Animation, pp. 306–312. Hyperion, New York (1995)

Vilhjálmsson, H.H.: Augmenting online conversation through automated discourse tagging. In: Proceedings of the 38th Annual Hawaii International Conference on Systems Science System Sciences, HICSS 2005, pp. 109a-109a. IEEE, January 2005