

Trade-offs between Responsiveness and Naturalness for Player Characters

Aline Normoyle*
University of Pennsylvania

Sophie Jörg†
Clemson University

Abstract

Real-time animation controllers are fundamental for animating characters in response to player input. However, the design of such controllers requires making trade-offs between the naturalness of the character's motions and the promptness of the character's response. In this paper, we investigate the effects of such trade-offs on the players' enjoyment, control, satisfaction, and opinion of the character in a simple platform game. In our first experiment, we compare three controllers having the same responsiveness, but varying levels of naturalness. In the second experiment, we compare three controllers having increasing realism but at the expense of decreased responsiveness. Not surprisingly, our least responsive controller negatively affects players' performance and perceived ability to control the character. However, we also find that players are most satisfied with their own performance using our least natural controller, in which the character moves around the environment in a static pose; that differences in animation can significantly alter players' enjoyment with responsiveness being equal; and that players do not report increased motion quality with our most natural controller, despite viewers outside of a game context rating the same controller as significantly more natural than our other conditions.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation; K.8 [Personal Computing]: General—Games

Keywords: digital games, controller, virtual character, perception, motion quality, responsiveness

1 Introduction

Animation techniques vary in the level of realism and amount of responsiveness they provide. As animators, we often operate under the assumption that greater naturalness, defined as *how realistic is a virtual character's performance*, improves the player's experience.

In this work, we evaluate this premise in the context of player controls for video games, where the desire for motion naturalness competes with a need for high responsiveness. In particular, poor responsiveness can interfere with the player's ability to play and can thus lead to frustration and decreased enjoyment as well as

* e-mail: alinen@seas.upenn.edu

† e-mail: sjoerg@clemson.edu

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or to publish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from Permissions@acm.org.

MIG '14, November 06 - 08 2014, Playa Vista, CA, USA
Copyright 2014 ACM 978-1-4503-2623-0/14/11\$15.00
<http://dx.doi.org/10.1145/2668064.2668087>

decreased performance [Quax et al. 2004; Claypool and Claypool 2006; Jörg et al. 2012; Amin et al. 2013; Normoyle et al. 2014].

In general, *responsiveness* refers to the time between the input of the user and the time when the user sees a reaction of the character. Much previous work has investigated the impact of delays on players, particularly in the domain of networked multiplayer games, where such delays are typically unavoidable. However, unlike previous work, which studies delays between the user's input and the processing of that input (for example, caused by network latency), we study delays introduced by the animation system. In particular, we are interested in how much delay might be tolerated by the player when it allows the character to move in a more natural way.

As an example, let's consider the scenario of a real person responding to a command to jump. Even if the person's reaction is immediate, if the request occurs when the left foot is about to hit the ground, the person has to slightly bend its left knee, lower its center of mass, and gather momentum. The jump then is carried out with the next step. The time between the request and the actual action might be a few hundred milliseconds. Such a slow response might annoy players because it makes the game more difficult to play. In a virtual environment, where expectations on realism are possibly different, the player may prefer that the character abruptly switches to a jump animation despite discontinuities in the motion.

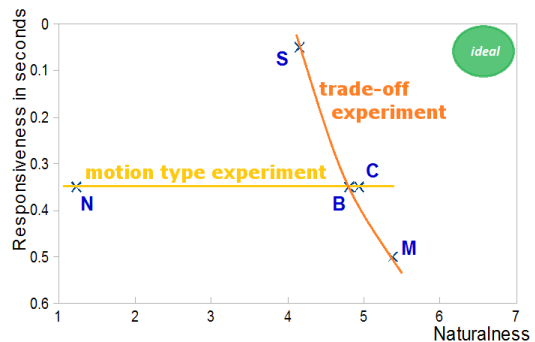


Figure 1: Overview of our controllers: NoMotion, Blend, and Cape, which all have the same responsiveness, are examined in the motion type experiment while Switch, Blend, and MoveTree serve to draw conclusions in the trade-off experiment. Values for responsiveness were obtained via simulation of user input (Section 4.7). Values for naturalness were obtained from an additional experiment which asked participants to specify the naturalness of animations created with each controller (Section 4.8). Each controller is represented with its first letter (Section 3.2). The green area in the top right shows the theoretically ideal controller with highly natural motions and a nearly immediate response.

We define delays due to the animation system as *animation responsiveness*. Animation responsiveness is the time between when the application receives input and when the character completes the request in a way that is visible to the user. We designate the algorithm used to animate a character based on the input of the user and the reactions of the character to the game world the *animation controller* or *character controller*. A typical animation controller's goal is to switch between behaviors, for example to let a jogging character jump when the user requests it. Does the naturalness of motions

significantly improve the player’s experience or is a quick animation response time more crucial for the player? By understanding the trade-offs between animation responsiveness and naturalness, we can ensure that animation complexity is only added where it enhances the player’s enjoyment of the game.

The ideal method for answering the above questions is to create two series of experiments. In the first, the naturalness remains constant while responsiveness varies and in the second, the responsiveness remains constant but the naturalness varies. The first series has already been investigated by Normoyle et al. [2014]. However, for the reasons outlined earlier, creating the second series of experiments is infeasible since it is not possible to maintain high realism while also keeping responsiveness high.

We therefore use the following approach: we design two experiments with three controllers each. The first, called the *motion type* experiment, investigates the effect of changing the animation of the character while leaving the responsiveness constant. We evaluate three controllers: NoMotion, in which the character body remains in a fixed pose; Blend, in which the character is animated and smoothly transitions in response to player input; and Cape, which is animated in the same way as Blend, but adds an animated cape to give secondary motion cues. The second, called the *trade-off* experiment, investigates the effect of reducing the responsiveness of the character in exchange for additional naturalness. We evaluate three controllers: Switch, in which the character transitions immediately with abrupt transitions and hence has the fastest responsiveness; Blend, which is the same as in the first experiment and thus has slower responsiveness than Switch but supports smooth transitions; and MoveTree, which contains several transitions that are much slower than in the Blend condition in order to support more realistic turning. Figure 1 gives an overview of our controller attributes.

Using these controllers, we investigate the impact of animation response time and naturalness on the player’s experience of a simple platform game. Our participants only experience one type of controller and are not aware of the purpose of our experiment. We find that in our setting a quick responsiveness is more important than a realistic animation. Our least responsive controller both negatively affects players’ performance and their perceived ability to control the character. However, we also find

- that players are most satisfied with their own performance using our least natural controller, in which the character moves around the environment in a static pose,
- that differences in animation can significantly alter players’ enjoyment with responsiveness being equal, and
- that players do not report increased motion quality with our most natural controller, despite viewers outside of a game context rating the same controller as having significantly more natural motions than our other conditions.

In Section 2, we present related work and in Section 3, we present the design of our experiment. In Section 4, we describe and analyze our controllers and their responsiveness and naturalness in detail. Finally, in Section 5, we present and discuss the results of our experiments, followed by Section 6 where we discuss conclusions and reflections on future work.

2 Related Work

2.1 Animation Controllers

Van Welbergen et al. [2010] give a good review of modern character animation techniques rated based on naturalness (defined as

the perceived realism of the character’s movements) and on control (defined in terms of responsiveness, precision, coverage, expressiveness, and intuitiveness). With their categorization, procedural techniques have the highest potential for control but the lowest naturalness whereas motion capture has the least control but the highest naturalness.

For player avatars, animation controllers must support interactive control as well as look believable. In the game industry, a popular way to create such animation controllers is with a carefully assembled move tree [Kines 2000; Lau and Kuffner 2005; Unity 3D Manual 2014]. These move trees are finite state machines, where each state is a motion clip, for example, a walk cycle or a jump, and transitions between states are motion transitions. Because techniques based purely on blending tend not to scale (often growing exponentially in practice [Gleicher 2008]), such move trees are often combined with procedural techniques to support interacting with a dynamic game environment, for example, inverse kinematics to support reaching, rag doll physics to respond to impacts, and biologically-based gaze models to support eye movements.

Because the manual design and creation of move trees is very time consuming, research has investigated methods to automate the process. Motion graphs were developed as a technique to automatically generate move trees from large motion capture databases [Arikan and Forsyth 2002; Kovar et al. 2002; Lee et al. 2002] which can then be postprocessed to create interactive controllers [Lee and Lee 2006]. Alternatively, databases of motion clips can be automatically assembled into interactive control policies directly [Gleicher et al. 2003; Treuille et al. 2007; McCann and Pollard 2007; Lee et al. 2010]. Interestingly, McCann and Pollard [2007] assembled very short motion fragments (0.1s each) on the fly and showed that occasional low-quality transitions improve overall motion quality.

An intrinsic issue with many automatic approaches, such as motion graphs, is the trade-offs between responsiveness, smoothness, and realism: if the chosen threshold for generating transitions is too high, the quality of the motion transitions suffers and the transitions look unnatural. If the threshold is too low, the connectivity of the graph suffers, and the time required to get to a specific animation increases, which results in a poor responsiveness. Additionally, automatic approaches often suffer from poor coverage as it is hard to know in advance whether the constructed controller supports the application needs. Reitsma and Pollard [2007] proposed a method for automatically evaluating the coverage of a motion graph. Cooper et al. [2007] developed a technique for suggesting additional motions to capture to improve the coverage of the controller. To further address such limitations, research has proposed numerous enhancements and alternative approaches to motion graphs [Heck and Gleicher 2007; Arikan and Forsyth 2002; Zhao and Safonova 2008; Chai and Hodgins 2005; Lee et al. 2010]. However, even as automatic techniques improve, such methods tend to remain memory intensive, requiring many example motions and a lot of memory to store the corresponding interactive controller. Additionally, automatic approaches do not address the problem of responsiveness versus naturalness: the responsiveness expected from games often requires unrealistic movements, such as those that violate balance, result in large torques, or require instant changes in acceleration or center of mass trajectory while in air. To give a common example, a realistic transition will often require a character to take an extra step. For many games, such as those having platforms that require jumps, this extra step can kill the player character!

For this study, we develop controllers manually using combinations of blending, move trees, and inverse kinematics, as these are the most widely used in video games today and offer the highest degree of responsiveness. However, the results of this study are applicable to automatic methods as well. By investigating how natural-

ness and animation responsiveness are perceived by the player, we aim to give researchers and designers the ability to make informed decisions on what to prioritize and to contribute to the effort of developing even better animation controllers.

2.2 Perception of delays in games and animation

The impact of network delays on player performance has been studied extensively, particularly concerning thresholds for how much delay is tolerable by players. Claypool and Claypool [2006] identified tolerable levels of latency for different categories of games, for example, suggesting that fast paced, high precision actions, such as first person shooters and racing games, should have a maximum latency of 100ms whereas third-person games can tolerate delays of up to 500ms. In contrast, Dick et al. [2005], when comparing player performance in Unreal Tournament, Counter-Strike, and Need for Speed, demonstrated that although a 150ms delay was noticeable, a 500ms delay was still playable and did not decrease performance. Beigbender et al. [2004] found that latencies as low as 100ms can significantly degrade performance in Unreal Tournament 2003 and latencies over 150ms make the game feel sluggish. In Unreal Tournament 2003, Quax et al. [2004] found that jitter values up to 95ms did not impact players whereas latencies starting at 60ms did. For a 3D platform game, Jörg et al. [2012] found that an average of 150ms of delay significantly reduced the players perceived ease of control, their performance, as well as their satisfaction with it and Normoyle et al. [2014] found that when such delays were constant, only very large delays negatively affected enjoyment, frustration, and ease of control. Beznosyk et al. [2011] observed that in the cooperative, multiplayer game Little Big Planet 2 delays over 100ms significantly decrease performance. Amin et al. [2013] observed a drop in their average ratings between delay jitters of 100ms and 250ms for subjects playing Call of Duty Modern Warfare 2.

We are not aware of research on the impact of motion realism in video games. However, numerous studies investigate the influence of motion realism in animated clips and movies. Several studies investigate the concept of the Uncanny Valley that hypothesizes that our familiarity or emotional engagement with a robot or virtual character increases with its realism, except for a sharp drop for entities that are close to but not fully human like, an effect that increases when motion is involved [Mori 1970; Hodgins et al. 2010]. Other studies show that we are very good recognizing and interpreting details in human motions. For example, we can determine the gender and even the individual above chance based on a few points attached to the body [Cutting and Kozlowski 1977], recognize emotions based on body motions or even posture [Atkinson et al. 2004], and detect even small errors in animation [Hoyet et al. 2012; Jörg et al. 2010; Trutoiu et al. 2011]. Previous research by McDonnell et al. [2007] investigated thresholds for motion smoothness and found that motions with lower complexity and linear velocity needed significantly fewer pose updates. Although much evidence supports the notion that people are very good at recognizing and interpreting even small details in human motions, it is unclear whether this holds in a video game setting where the player is not necessarily focused on the motions of the game character, but instead on reaching game-related goals.

3 Design

We investigate the impacts of naturalness and responsiveness on the player’s experience, closely following Jörg et al.’s [2012] experimental method.

3.1 Hypotheses

Previous work has shown that large latencies reduce players’ performance (see Section 2), and can additionally reduce players’ enjoyment of a game [Jörg et al. 2012] as well as the perceived motion quality of the avatar [Normoyle et al. 2014]. Unlike such previous work, where a delay is introduced in processing the user’s input, we process all user input immediately, but investigate delays introduced by the animation system. Specifically, we look at delays inherent in transitions between avatar behaviors, for example, due to crossfading between motions or by using realistic turns. In Section 4, we describe in detail the type of variation in responsiveness that different controller implementations can introduce.

In general, we expect a higher naturalness to have a positive impact on the enjoyment of a player, but that it will not affect the ability of the user to control the character or his or her performance. However, we only expect this effect until the responsiveness becomes too slow or too inconsistent (e.g. when some transitions are much slower than others).

Index	Hypothesis	Confirmed?
H1 Control		
H1-a	The player finds it more difficult to control the character when the responsiveness becomes very slow or inconsistent (MoveTree controller).	Yes
H1-b	Small delays in responsiveness do not affect the player’s perceived ability to control a character. The naturalness of the motions do not affect the player’s perceived ability to control the character.	Yes
H2 Enjoyment		
H2-a	The player’s enjoyment is not affected by the responsiveness of the controller (based on the results from Normoyle et al. [2014]).	Yes
H2-b	The player’s enjoyment increases when naturalness increases.	No
H3 Satisfaction		
H3-a	The player’s satisfaction increases as responsiveness increases.	No
H3-b	The player’s satisfaction is unaffected by naturalness.	No
H4 Performance		
H4-a	The player’s performance decreases as responsiveness decreases below a threshold.	Yes
H4-b	The player’s performance is unaffected by naturalness.	Yes
H5 Perception of character and game		
H5-a	The player’s perception of the character and the game will not change as responsiveness decreases, except for the perceived motion quality of the character, which will decrease (based on the results from Normoyle et al. [2014])	No
H5-b	The player’s perception of the character and the game will be more favorable when the naturalness increases.	No

Table 1: Summary of our hypotheses and findings.

3.2 Method

We design two experiments to evaluate trade-offs between naturalness and responsiveness. In the first, we compare three controllers having the same level of responsiveness, but differing levels of naturalness. In the second experiment, we compare three controllers having increasing realism but at the expense of decreased responsiveness (see Figure 1).

For each experiment condition, we use the game developed by Jörg et al. [2012] and only change the character controller (see the accompanying video for examples of the stimuli and game environment). This game consists of three levels, each implemented using the Unity [Unity 3D Manual 2014] game engine. To investigate possible learning effects, level 1 and level 3 are exactly the same and consist of a path for participants to follow with their character. Level 2 is intended to be more difficult and consists of elevated platforms from which the player can die if they do not time jumps correctly. In every level, players must avoid lasers to not lose health and can optionally collect gems.

The player traverses the environment using a gamepad to steer a female, human like character who can stand, jump, and jog. A third-person camera automatically follows the character (chosen to accommodate players with less game experience who would find simultaneously steering the character and camera too difficult). The game starts in a tutorial area where the player can become accustomed to the controls before starting the experiment.

Each controller is based on the same three primary motion states, jumping, jogging, and standing. The animations for each state were obtained from motion capture. The player input used to control the character is identical in every condition. Players use the left analog stick to steer the character. Without left analog input, the character stands idly. Pressing the gamepad’s X button triggers a jump.

We implement four controllers of varying naturalness and responsiveness.

- Switch (S): The character transitions immediately between motions without blending. This condition is the most responsive, but the animation is discontinuous when transitioning.
- Blend (B): The character smoothly blends between motions and turns. Transitions occur as quickly as for the switch condition, but because of crossfading over several frames, the change is not immediately visible to the player.
- MoveTree (M): The character smoothly blends between motions through crossfading as well as supports realistic turns based on motion capture data (implemented using blend trees). Our aim for this condition was a very high naturalness without decreasing the responsiveness so much as to hamper gameplay. Although this controller has several faster transitions than the Blend controller, this condition has the least responsiveness overall because turning is slower and it is not possible to switch mid-transition.
- NoMotion (N): The character’s body is not animated. The avatar’s position and orientation are updated in the same way as the Blend controller but the pose remains static. This condition has the same responsiveness as the Blend controller and the lowest naturalness of all conditions.
- Cape (C): To potentially add realism through secondary motion cues, we add an animated waist-length cape to the blend controller condition. The cape moves in a physically-plausible way using mass-spring dynamics. This controller has the same responsiveness and character animation as the Blend controller, but alters the character appearance.

Details about the implementation of each controller type and an analysis of its responsiveness and motion naturalness are given in Section 4.

3.3 Participants

We recruited participants through flyers and class announcements. In total 67 participants, mostly students, from a variety of majors participated in this study. Each participant received \$5 at the end of the experiment. The number of participants in each condition is listed in the following table.

condition	participants	male/female
N	13	8m, 5f
B	13	8m, 5f
C	10	6m, 4f
S	18	10m, 8f
M	13	8m, 5f

3.4 Procedure

We used a between-group design so that users could not compare different conditions. Each participant played through all three levels with the same animation controller.

Before beginning, participants were given written information regarding the study, and were asked to provide their gender, age, and experience with digital games and virtual characters. Next, they were provided instructions describing the controls and were asked to play through all three levels. Before starting each level, participants received verbal instructions for each level. In the identical levels 1 and 3, participants were asked to follow a path as quickly and as accurately as possible. In level 2, they were asked to reach the big gem at the opposite end of the environment.

During game play, we log several metrics to gauge the players performance, such as the number of gems collected, the amount of health and lives lost, and the time needed to reach the goal. After completing each level, participants were asked to answer four questions on 7-point Likert scales specific to the level last played: How much did you enjoy playing this level?, How did you feel about your performance in this level?, How difficult/ easy was it to control the character in this level?, and How would you rate the quality of the motions in this level?. After completing all levels, participants were additionally asked to rate the game (e.g., how entertaining, challenging, or repetitive it was), rate the character (e.g., how agile, human like, or sympathetic it was), and to give reasons why they rated the animations as high or low quality, why they liked or disliked the game, and what they thought the purpose of the study was. They were also given time to write down any comments they may have. Afterwards, the purpose of the study was explained to them and they were asked to sign a consent form so that we may use their data. On average, the study required 30 minutes to complete.

4 Character controllers

Motion capture remains the most common method for animating human like virtual characters in current games [Gleicher 2008]. We therefore use motion capture as the basis for the character controllers created for this study. Each controller was implemented in the professional game engine Unity [Unity 3D Manual 2014].

These controllers consist of stand, jog, turn, and jump motions of a female performer. All input motions were at 30 fps. Each motion was trimmed to start and end on a left foot contact (with the exception of standing). The stand and jog motions were preprocessed to loop.

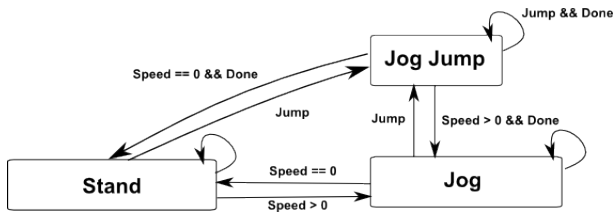


Figure 2: Finite state machine used for the NoMotion, Switch, Blend, and Cape conditions. Note that standing and jogging automatically loop where jumping only repeats based on boolean state variables.

4.1 Switch (S) Condition

In the Switch (S) condition, the character switches immediately between animations with no blending according to the state machine in Figure 2. When jogging or standing, heading is smoothly interpolated based on user input and foot contacts are aligned using inverse kinematics to prevent foot sliding. However, discontinuities in both the feet and character pose occur for large turns and when switching between standing, jogging, or jumping.

4.2 Blend (B) Condition

The Blend (B) condition crossfades smoothly between motions using the same state machine as the Switch (S) condition (Figure 2). Same as Switch (S), the heading is smoothly interpolated and foot contacts are aligned using inverse kinematics. In this condition, the character responds immediately to user input, although the transition may not be immediately visible to players because of the blending. Transitions can occur at anytime from the source motion. To align feet contacts during blending, we dynamically select the best match frame from the target motion for blending. Additionally, players can transition to a new motion even while transitioning.

4.3 MoveTree (M) Condition

In the MoveTree (M) condition, the character performs animated jumps, jogs, and stands with realistic turning animations captured from the female performer based on the state machine in Figure 3. This controller was created using Unity 4.0's Mecanim tools. The jogging subcontroller consists of 5 input motions: 2 left turns, jog forward, and 2 right turns (Figure 4). The standing subcontroller consists of 3 input motions: one left turn, stand, one right turn (Figure 4). Turning motions are then computed by computing a blend weight between two input motions based on the desired relative heading of the character, e.g. when the controller direction is 45°, the character is animated using a 50% blend between the forward jog and the 90° turn. Additionally, we include a realistic 180° turn which is triggered during large heading changes. Lastly, the straight-line jog, stand, and jogging jump used for this controller are the same as the ones used in the other three controllers, but we use an additional captured motion for jumping from standing.

Because this controller is intended to be realistic, the character can change heading only through the captured turn animations. This is unlike the previous controllers where the character heading is directly set to match the desired heading (or blended towards the desired character heading). Additionally, it is not possible to transition from a transition, nor to turn while in air jumping. Lastly, the character speed is smoothly increased when transitioning from standing to jogging. This allows the character to turn in place while standing.

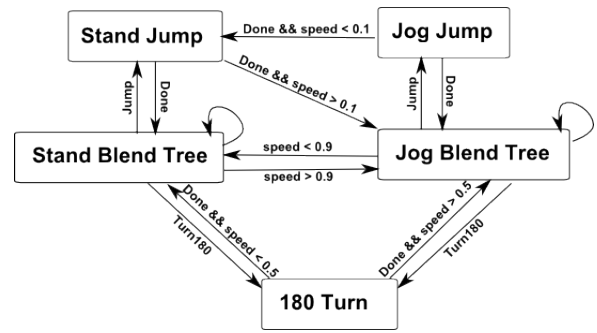


Figure 3: Finite state machine used for the Move Tree condition.

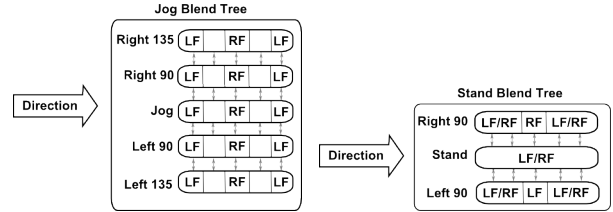


Figure 4: Blend Trees used for jogging (left) and standing (right) in the MoveTree (M) condition.

4.4 NoMotion (N) Condition

In the NoMotion (N) condition, the character position and heading are updated each frame in the same way as the Blend (B) controller (Figure 2) except that the character pose now remains fixed in a static posture (Figure 5). In this condition, the character responds immediately to user input but similarly to the Blend (B) condition, the transition may not be immediately visible to the viewer.



Figure 5: Left, a pose from the jogging motion used for our controllers. Right, in the NoMotion condition, the character moves through the environment in a static posture.

4.5 Cape (C) Condition

In the Cape (C) condition, the character is animated using the same controller as the Blend (B) condition but is modified to have an animated waist-length cape. The responsiveness of this controller is identical to the Blend (B) and NoMotion (N) conditions.

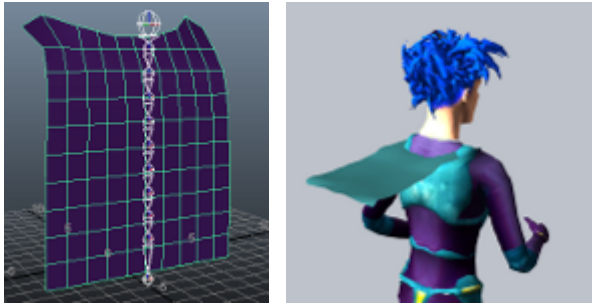


Figure 6: In the Cape (C) condition, we add an animated cape to the Blend controller. Left, the cape consists of a skinned mesh controlled by a single chain of control bones. Right, the cape is rigidly connected to the character’s shoulders. Physically-plausible movement is achieved by animating the chain of control bones as a mass spring system.

The cape is animated as a skinned mesh controlled by a single chain bone skeleton. To create physically-plausible cape movements in response to the character’s movement, we simulate the bones of the cape skeleton as a mass-spring system rigidly connected to the character at the shoulders (see Figure 6). This simplified cloth implementation is very fast as well as stable in response to the large forces generated by the character during movement.

4.6 Summary of differences between controller responsiveness

Many of the design decisions for each controller directly affect its responsiveness.

Turning:

- Move Tree (M): Because all turns are based on captured data, the turning speed is limited by the captured motions. Consequently, the character can not turn as tightly as in the other conditions and requires more time to make a large turn. All turns are smooth.
- Switch (S): The character heading is set immediately based on user input. Although there is no delay, the animation is discontinuous.
- Blend (B)/NoMotion (N)/Cape (C): The character blends towards the player’s desired heading. The maximum turn angle is 10° per frame. Thus, larger turns take more time than smaller turns, but the resulting animation is smooth.

Jumping:

- MoveTree (M): Can not turn while jumping, nor can transition behaviors mid-jump.
- Switch (S)/Blend (B)/NoMotion (N)/Cape (C): Can turn while jumping, but can not transition behaviors mid-jump.

Transitions:

- MoveTree (M): With the exception of transitioning from stand to jog (where the speed is smoothly ramped up), all transition times are set up a priori and take a fixed amount of time ($\approx 0.2s$). Although it is not possible to transition while transitioning, the character can initiate a transition from any point while standing or jogging.

- Switch (S): The character immediately switches behavior. There is no delay, but the motion is discontinuous.
- Blend (B)/NoMotion (N)/Cape (C): The character immediately changes behavior and can transition from a transition. The source and target behaviors are crossfaded over 10 animation frames, meaning the player may not see the result of the transition immediately.

4.7 Responsiveness of controllers

We compute the average responsiveness of each controller by simulating the user input which triggers transitions between jumping, standing, jogging, and large turns (Table 2). The goal of this analysis is to determine the length of animation transitions in response to the player’s input.

Transition	Switch	Blend/NoMotion/Cape	Move Tree
Stand to Jump	0.03	0.37	0.1
Stand to Jog	0.03	0.36	1.3
Jog to Jump	0.02	0.33	0.23
Jog to Stand	0.03	0.36	0.24
Turn 90	0.03	0.31	1.41

Table 2: Simulated transition and turning times in seconds.

Times were collected on an Intel Core2 Duo CPU 2.10 GHz, 4 GB RAM, 32-bit OS, where the application frame time was approximately 0.005s. Thus, all transitions fluctuate by 15ms depending on the timing of input. At 30 fps, the animation frame rate is approximately 0.033s. The Blend, Cape, Switch, and NoMotion controllers update the character pose at the animation frame rate. For this reason, the Switch condition (which transitions immediately) takes on average one animation frame to transition to the next behavior. This is the minimum possible. The Blend transition times include the 10 frames used to cross fade, which at 30 fps corresponds to $0.03 * 10 = 0.3s$ for the entire crossfade to occur. In regards to responsiveness, the blend condition likely appears more responsive than these numbers suggest, since the player likely sees that the character has transitioned to the new motion before the crossfade ends. Lastly, in the Move Tree the delay is determined by the transition lengths which were setup a priori. Note that the transition from stand to jog is longer because we gradually increase the character speed from standing to jogging to support a standing turn.

We also simulate a 90° turn for each controller. For these simulations, we hard code the joystick input to a 90° turn and log the amount of time until the character is within 10° of the desired heading. The Switch (S) condition changes immediately and has the same amount of delay as for switching behaviors. The Blend (B)/NoMotion (N)/Cape (C) conditions smoothly turns the character at 30 fps with a maximum angle of 10° per frame. Thus, a 90° turn takes minimally $90/10 * 0.03 = 0.27s$ to get within 10° of the target heading (and a little longer in practice). The MoveTree (M) condition, which uses a blend tree of captured turns, takes the longest with 1.4s.

4.8 Naturalness of controllers

To determine the perceived naturalness of each controller, we design an experiment in which we ask participants to rate the naturalness of animations captured from each controller type.

Unlike the game experiment, where participants only see one controller type, participants in this experiment see animations created with each controller type. We created five animations for each controller: jogging, jumping from standing, jumping twice from jog-

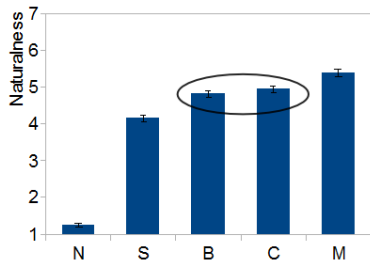


Figure 7: Averaged ratings for the naturalness of the motions in each controller. All differences are significant except for the controllers Blend and Cape. Error bars represent one standard error of the mean in all graphs.

ging, turning 90° , and turning 180° . All animations were captured in the tutorial area of level 1. Using an online survey, participants saw the 25 clips in random order repeated three times, for a total of 75 clips in each survey.

The 12 participants were students and faculty recruited by e-mail ranging from ages 19 to 34, 7 male, 5 female. We averaged the ratings over the three repetitions and five examples for each controller and participant. A repeated measures ANOVA showed significant differences between the five controllers with $F(4, 44)=131.7$ and $p<0.001$. A Newman-Keuls post hoc test shows that the differences between all controllers are significant except for the Blend and Cape controller. The NoMotion controller was rated as least natural by far, which is not surprising as the character does not move. Rated most natural were the motions in the MoveTree controller, followed by the motions in Blend and Cape, and then the motions in the Switch controller. The results correspond exactly to our expectations and can be seen in Figure 7 and Figure 1.

5 Results and Discussion

Two-way repeated measure ANOVAs with the factors Controller and Level were used to analyze the responses to the questions about enjoyment, control, satisfaction, and motion quality that we asked after each level as well as the collected performance metrics when more than one level was considered. Controller is a between-subjects variable with the three values NoMotion, Blend, and Cape in the motion type experiment and Switch, Blend, and MoveTree in the trade-off experiment, respectively. Level is a within-subjects variable with three values, representing the three levels that were played, in each of the experiments. Level has only two values in the analysis of performance metrics when only level 1 and 3 were used in the analysis. When significant effects were present, we used Newman-Keuls post hoc tests to determine their reasons.

We analyzed the answers about realism and about the game and character attributes with one way ANOVAs with the factor Controller with the same values than in the two-way ANOVAs.

Overviews of our results are shown in Tables 1 and 3. Representations of the main results can be seen in Figures 8, 9, and 10.

5.1 Control

There is no effect of Controller ($F(2,33)=0.76$, $p\approx 0.48$) on the perceived ease of control in the motion type experiment, which means that the perceived ease of control does not change when only the motions but not the responsiveness are varied. That corresponds to our expectations. In the trade-off experiment, there is a significant effect of Controller ($F(2,41)=3.59$, $p<0.05$). A Newman-Keuls post

hoc test reveals that this is due to the MoveTree condition, where ease of control was rated lower than the Switch condition and nearly lower than the Blend condition ($p<0.1$). We attribute this result to the fact that the responsiveness of the MoveTree controller is slower than the responsiveness of Blend and Switch.

Based on those results, we can support H1-a and H1-b. We conclude that the naturalness does not affect the players' perceived ease of control. A low responsiveness, however, negatively affects the perceived ease of control and a higher naturalness can not compensate for that effect.

We also find a main effect of Level with the character considered hardest to control in level 2 and easiest in level 1 with all differences between levels being significant in both experiments.

5.2 Enjoyment

When examining the enjoyment ratings, we find a significant effect of Controller in our motion type experiment with $F(2,33)=3.37$ and $p<0.05$. The condition with the cape was enjoyed significantly less than the Blend condition. As the only difference between those two controllers is the added cape, which is a visual element that does not alter the responsiveness, we infer that such extra motions can influence the player enjoyment. There is no effect in our trade-off experiment ($F(2,41)=1.60$, $p\approx 0.21$).

We confirm hypothesis H2-a, that the player's enjoyment is not affected by the responsiveness of the controller. But we can not support H2-b based on these results. We find that added motion effects can alter the players enjoyment. However, we do not have any evidence that relates that change to naturalness.

Again, there is a main effect of Level, this time with the second level being enjoyed significantly more than the first and third levels in both experiments.

5.3 Satisfaction

In the motion type experiment, the answers to the question how satisfied participants felt with their performance after each level show an effect of Controller significant at the 0.1 level with $F(2,33)=2.77$ and $p<0.1$ due to the fact that participants felt more satisfied after playing with the controller without any body motions (N) than after playing in the other two conditions (B and C). Participants might have thought that this condition was particularly easy and therefore felt more satisfied with their performance on average.

There is no effect of Controller in the trade-off experiment ($F(2,41)=0.24$, $p\approx 0.79$). Similar to the other answers, we found a main effect of Level. Participants are most satisfied with their performance in level 3 in both experiments. In the trade-off experiment players are also significantly more satisfied after playing level 1 than after playing level 2, while this difference is not significant in the motion type experiment.

We can not confirm hypotheses H3-a nor H3-b, that the player's satisfaction increases as responsiveness increases and is unaffected by naturalness. A higher naturalness might decrease the player's satisfaction since satisfaction values were lower for our Blend (B) and Cape (C) conditions.

5.4 Performance

We analyze several performance metrics, such as the number of gems collected and the time to reach the goal in level 1 and 3 and the number of lives lost in level 2. Level 2 was excluded from some

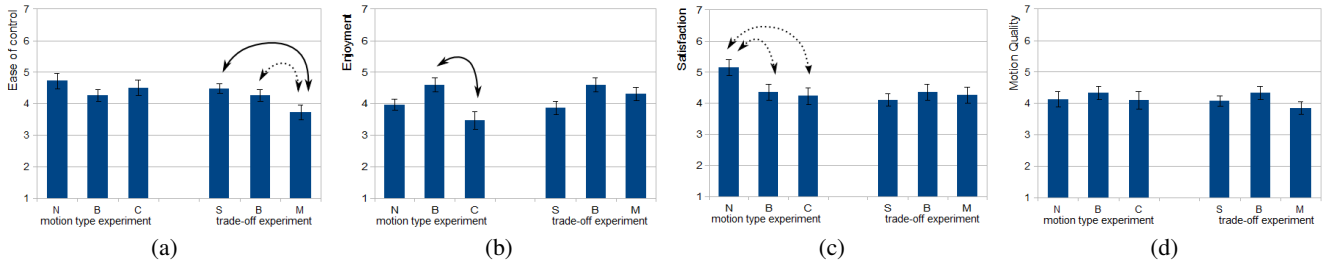


Figure 8: Average ratings for the controllers in both experiments for the questions on (a) Ease of control, (b) Enjoyment, (c) Satisfaction, and (d) Motion quality. Significant effects were found for the ratings of the ease of control in the trade-off experiment where $M < S$ and B , for the enjoyment ratings in the motion type experiment where $C < B$, and for the satisfaction ratings where $N > B$ and C . Arrows highlight significant differences with dashed arrows being used for differences significant only at the 0.1 significance level.

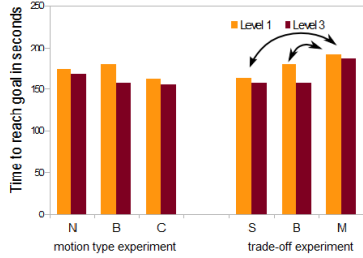


Figure 9: Time to reach goal in levels 1 and 3. There are significant main effects of Controller and Level for the trade-off experiment with $S, B < M$ and $L1 > L3$, but no significant effects for the motion type experiment.

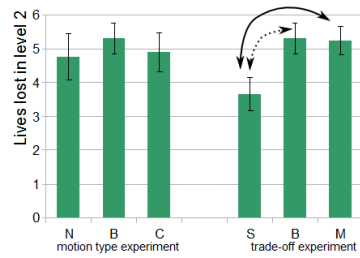


Figure 10: Lives lost in level 2. There is a significant effect for the trade-off experiment, but not for the motion type experiment.

metrics as too many participants did not reach the goal and therefore would, for example, also not have the opportunity to collect gems.

There are no significant differences of Controller when examining the number of gems collected in levels 1 and 3 in both experiments.

The time to reach the goal in levels 1 and 3 gives us interesting insights. There is no effect of Controller for the motion type experiment but there is a main effect of Controller in the trade-off experiment with $F(2,40)=10.58$ and $p < 0.001$ due to the fact that participants take significantly more time to reach the goal in the MoveTree condition than with the Switch or Blend controller. The results are displayed in Figure 9. There is also a significant effect of Level ($F(1,40)=5.09$, $p < 0.05$) in the trade-off experiment. Participants reach the goal in a shorter time in level 3 than in level 1, suggesting that participants' performance improved over time.

When considering the number of lives lost in the more challenging second level with a one-way ANOVA, we find a significant effect of Controller in the trade-off experiment ($F(2,40)=4.23$, $p < 0.05$) but not in the motion type experiment ($F(2,32)=0.14$, $p \approx 0.87$). As can be seen in Figure 10 when players use the Switch controller they use significantly less lives than in the MoveTree controller and the difference to the Blend controller is nearly significant ($p \approx 0.053$). This result might be due to the fact that the Switch controller has the highest responsiveness.

We conclude based on those results that the player performance does decrease for some of the analyzed metrics for the MoveTree condition (the time to reach the goal and the number of lives lost) and for one metric for the Blend condition (the number of lives lost). We can therefore confirm H4-a, that the player's performance decreases when the responsiveness is low, and H4-b, that the player's performance is not affected by naturalness.

	motion type	trade-off
control	—	$M < S^{**}, M < B^*$
enjoyment	$C < B^{**}$	—
satisfaction	$N > B, C^*$	—
motion quality	—	—
realism	—	—
game and character attributes	—	—
gems collected (L1,L3)	—	—
time to reach goal (L1,L3)	—	$S, B < M^{**}$
lives lost (L2)	—	$S < M^{**}, S < B^*$

Table 3: Overview of significant main effects for the factor Controller with $**$ to denote a significant effect with $p < 0.05$, $*$ when $0.05 < p < 0.1$, and $—$ for no effect.

5.5 Perception of the Character and the Game

There are no significant effects for any of the questions we asked about the perception of the character or the game, not for the motion quality in the motion type experiment ($F(2,33)=1.33$, $p \approx 0.88$) or in the trade-off experiment ($F(2,41)=0.68$, $p \approx 0.51$) nor for realism of the character or its human likeness in any experiment (all $p > 0.47$). We therefore can not confirm H5-a or H5-b.

Normoyle et al. [2014] found that a lower responsiveness, be it due to a delay or due to jitter, decreases the perceived motion quality. Based on participants' comments, they concluded that the ease of control was taken into account when rating the motion quality. Therefore, in the trade-off experiment one explanation for not finding any effect could be that the low responsiveness in the MoveTree condition did in fact reduce the perceived motion quality or realism as predicted but that the higher naturalness countered and compensated this effect. However, as we do not find any effects in the motion type experiments neither this is just speculation.

The absence of any effects in the motion type experiment is particularly hard to explain. Of course, our participants only see one specific condition and they might think that an immobile character that just floats around is part of the game concept. Furthermore, players' expectations are different when in a university experiment setting than when they play a commercial game. However, these arguments could be claimed for any of our conditions or questions and do not satisfactorily explain our results.

For the question on motion quality, there is a significant effect of Level with the motion quality in level 3 being perceived significantly higher than in both other levels in both experiments. Furthermore, in the trade-off experiment we find a significant interaction effect between Controller and Level. The Newman-Keuls post hoc test reveals that in the Blend condition, the motion quality was perceived to be significantly higher in level 3 than in level 2. Furthermore, in the Switch condition, the motion quality was rated higher in level 3 than in level 1.

6 Conclusion and Future Work

In this work, we investigate how trade-offs between naturalness and responsiveness can affect players' control, enjoyment, satisfaction, performance, and motion quality in a video game. Overall, our results show that reaching the required behavior quickly remains crucial in a game setting. Similarly to previous work, high responsiveness increases a player's objective performance, as demonstrated by players in the Switch condition requiring significantly less time to reach the goal and losing significantly fewer lives than in our MoveTree condition. However, despite the fact that objective measures did not show a higher performance of players in the NoMotion condition compared to players in the Blend and Cape conditions, those players were significantly more satisfied with their own performance, suggesting that character animation can interfere with how players perceive their own performance. Furthermore, our motion type experiment suggests that the character animation has a significant effect on players' enjoyment, potentially both for better and for worse, as suggested by our Cape condition which was enjoyed less by participants than the Blend condition even if the only difference was the cape.

In this work, we restricted many of our controller abilities in favor of realism, particularly for turns in the MoveTree (M) condition. We could have improved the responsiveness of the MoveTree (M) controller by scaling the playback speed of our turning animations or allowing transitions from transitions. Such changes may have resulted in the MoveTree condition being similarly easy to control as the Switch and Blend conditions but would have reduced the realism of the motions. However, motion alterations such as faster motions or higher jumps might be believable in a video game setting. Future work will be required to find out what types of motions players consider natural in a video game context, which might also depend on the character and the game environment and genre.

The participants in the online survey were asked to rate the naturalness of the motions but our post-level questionnaire asked participants to rate the motion quality. Our post-level questionnaire was deliberately kept the same as in previous work [Normoyle et al. 2014]; however, based on the results of both these studies, we believe participants may have interpreted the terms motion quality and naturalness differently than we intended. Specifically, numerous participants from Normoyle et al. [2014], in which all controllers were based on the switch controller, mentioned that their ability to control the character factored heavily into their ratings of motion quality.

Future work could investigate how well the effects discovered in this paper hold over longer play periods, or for different game types,

environments, and genres. Participants' previous experience with games might also influence their expectations and ratings. Also, even if naturalness may not be important for gameplay, high quality motions may help games sell due to their increased visual appeal, particularly for advertisements. Future work might also try to understand why and how animation might positively or negatively affect players' enjoyment or perceived control. For example, our participants enjoyed the cape controller significantly less, but we do not have insights into why this was so. Similarly, we do not know why players felt most satisfied with their performance for the NoMotion condition.

Although more investigation is necessary to fully understand the interaction between naturalness and responsiveness for player characters, our results strongly imply that responsiveness should always be prioritized over naturalness but suggest that the quality of the animation influences players' enjoyment.

Acknowledgments

We wish to thank our participants for their time, our reviewers for their comments, and Alla Safonova for early ideas and discussions regarding this work.

References

- AMIN, R., JACKSON, F., GILBERT, J. E., MARTIN, J., AND SHAW, T. 2013. Assessing the impact of latency and jitter on the perceived quality of call of duty modern warfare 2. In *Human-Computer Interaction: Users and Contexts of Use*, HCI'13, 97–106.
- ARIKAN, O., AND FORSYTH, D. A. 2002. Interactive motion generation from examples. *ACM Transactions on Graphics* 21, 3, 483–490.
- ATKINSON, A. P., DITTRICH, W. H., GEMMELL, A. J., AND YOUNG, A. W. 2004. Emotion perception from dynamic and static body expressions in point-light and full-light displays. *Perception* 33, 6, 717–746.
- BEIGBEDER, T., COUGHLAN, R., LUSHER, C., PLUNKETT, J., AGU, E., AND CLAYPOOL, M. 2004. The effects of loss and latency on user performance in Unreal Tournament 2003®. In *SIGCOMM Workshop on Network and System Support for Games*, NetGames '04, 144–151.
- BEZDOSYK, A., QUAX, P., CONINX, K., AND LAMOTTE, W. 2011. Influence of network delay and jitter on cooperation in multiplayer games. In *Virtual Reality Continuum and Its Applications in Industry*, ACM, 351–354.
- CHAI, J., AND HODGINS, J. K. 2005. Performance animation from low-dimensional control signals. *ACM Transactions on Graphics* 24, 686–696.
- CLAYPOOL, M., AND CLAYPOOL, K. 2006. Latency and player actions in online games. *Communications of the ACM* 49, 11, 40–45.
- COOPER, S., HERTZMANN, A., AND POPOVIĆ, Z. 2007. Active learning for real-time motion controllers. *ACM Transactions on Graphics* 26, 3.
- CUTTING, J. E., AND KOZLOWSKI, L. T. 1977. Recognizing friends by their walk: Gait perception without familiarity cues. *Bulletin of the Psychonomic Society* 9, 353–356.

- DICK, M., WELLNITZ, O., AND WOLF, L. 2005. Analysis of factors affecting players' performance and perception in multi-player games. In *SIGCOMM Workshop on Network and System Support for Games*, NetGames '05, 1–7.
- GLEICHER, M., SHIN, H. J., KOVAR, L., AND JEPSEN, A. 2003. Snap-together motion: Assembling run-time animations. *ACM Transactions on Graphics* 22, 3, 702–702.
- GLEICHER, M. 2008. More motion capture in games – can we make example-based approaches scale? In *Motion in Games*, MIG '08, 82–93.
- HECK, R., AND GLEICHER, M. 2007. Parametric motion graphs. In *Proceedings of the 2007 Symposium on Interactive 3D Graphics and Games*, I3D '07, 129–136.
- HODGINS, J., JÖRG, S., O'SULLIVAN, C., PARK, S. I., AND MAHLER, M. 2010. The saliency of anomalies in animated human characters. *ACM Transactions on Applied Perception* 7, 4, 22:1–22:14.
- HOYET, L., MCDONNELL, R., AND O'SULLIVAN, C. 2012. Push it real: Perceiving causality in virtual interactions. *ACM Transactions on Graphics* 31, 4, 90:1–90:9.
- JÖRG, S., HODGINS, J., AND O'SULLIVAN, C. 2010. The perception of finger motions. In *Applied Perception in Graphics and Visualization (APGV 2010)*, 129–133.
- JÖRG, S., NORMOYLE, A., AND SAFONOVA, A. 2012. How responsiveness affects players' perception in digital games. In *Symposium on Applied Perception*, SAP '12, 33–38.
- KINES, M., 2000. Planning and directing motion capture for games. http://www.gamasutra.com/view/feature/131827/planning_and_directing_motion_.php. [accessed September 2014].
- KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics* 21, 3, 473–482.
- LAU, M., AND KUFFNER, J. J. 2005. Behavior planning for character animation. In *Symposium on Computer Animation*, SCA '05, 271–280.
- LEE, J., AND LEE, K. H. 2006. Precomputing avatar behavior from human motion data. *Graphical Models* 68, 158–174.
- LEE, J., CHAI, J., REITSMA, P. S. A., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. In *SIGGRAPH*, 491–500.
- LEE, Y., WAMPLER, K., BERNSTEIN, G., POPOVIC, J., AND POPOVIC, Z. 2010. Motion fields for interactive character animation. *ACM Transactions on Graphics* 29, 138:1–138:8.
- MCCANN, J., AND POLLARD, N. 2007. Responsive characters from motion fragments. *ACM Transactions on Graphics* 26, 3, 6.
- MCDONNELL, R., NEWELL, F., AND O'SULLIVAN, C. 2007. Smooth movers: Perceptually guided human motion simulation. In *Symposium on Computer Animation*, SCA '07, 259–269.
- MORI, M. 1970. Bukimi no tani (the uncanny valley). *Energy* 7, 4, 33–35.
- NORMOYLE, A., GUERRERO, G., AND JÖRG, S. 2014. Player perception of delays and jitter in character responsiveness. In *ACM Symposium on Applied Perception*, SAP '14, 117–124.
- QUAX, P., MONSIEURS, P., LAMOTTE, W., DE VLEESCHAUWER, D., AND DEGRANDE, N. 2004. Objective and subjective evaluation of the influence of small amounts of delay and jitter on a recent first person shooter game. In *SIGCOMM Workshop on Network and System Support for Games*, NetGames '04, 152–156.
- REITSMA, P. S. A., AND POLLARD, N. S. 2007. Evaluating motion graphs for character animation. *ACM Transactions on Graphics* 26, 4.
- TREUILLE, A., LEE, Y., AND POPOVIĆ, Z. 2007. Near-optimal character animation with continuous control. In *ACM Transaction on Graphics*, vol. 26.
- TRUTOIU, L. C., CARTER, E. J., MATTHEWS, I., AND HODGINS, J. K. 2011. Modeling and animating eye blinks. *ACM Transactions on Applied Perception* 8, 3, 17:1–17:17.
- UNITY 3D MANUAL, 2014. Mecanim animation system. <http://docs.unity3d.com/Manual/MecanimAnimationSystem.html>. [accessed September 2014].
- VAN WELBERGEN, H., VAN BASTEN, B. J. H., EGGES, A., RUTKAY, Z. M., AND OVERMARS, M. H. 2010. Real time animation of virtual humans: A trade-off between naturalness and control. *Computer Graphics Forum* 29, 2530–2554.
- ZHAO, L., AND SAFONOVA, A. 2008. Achieving good connectivity in motion graphs. In *Symposium on Computer Animation*, no. 139–152 in SCA '09.