Adding Cartoon-like Motion to Realistic Animations

Rufino R. Ansara and Chris Joslin
School of Information Technology, Carleton University, 1125 Colonel By, Ottawa, Canada

Keywords: Computer Graphics, Cartoon, Motion Capture, Three-dimensional Graphics and Realism, Animation.

Abstract: In comparison to traditional animation techniques, motion capture allows animators to obtain a large amount of realistic data in little time. In contrast, classical animation requires a significant amount of manual labour. The motivation behind our research is to look at methods that can fill the gap that separates realistic motion from cartoon animation. With this knowledge, classical animators could produce animated movies and non-realistic video games in a shorter amount of time. To add cartoon-like qualities to realistic animations, we suggest an algorithm that changes the animation curves of motion capture data by modifying their local minima and maxima. We also propose a curve-based interface that allows users to quickly edit and visualize the changes applied to the animation. Through our user studies, we determine that the proposed curve interface is a good method of interaction. However, we find that in certain cases (both user-related and algorithmic), our animation results exhibit unwanted artefacts. Thus, we present various ways to reduce, avoid or eliminate these issues.

1 INTRODUCTION

Since the release of Toy Story, the first full-length computer-animated cartoon movie, cartoon animation has slowly transitioned from 2D to 3D animation methods. Studios like Pixar and DreamWorks use manually created character rigs, inverse kinematics and keyframe based animation to bring their characters to life – no motion capture is used. While these methods give the animator full control over the final animation, they also involve significant manual labour as animators must manipulate all the character's limbs independently.

Motion capture is significantly less time consuming than classical animation; although there might be more setup time required for the real-environment and post-processing of the animation itself; its emergence has revolutionized computer animation. This success is apparent in the dominance of high quality, realistic, animations used in movies such as Lord of the Rings, Avatar, and Planet of the Apes, and in games such as Final Fantasy, Grand Theft Auto, Call of Duty, and Halo.

However, despite the quality of the animations produced, motion capture is seldom, if ever, used by 3D animation studios for non-realistic characters. These studios adhere to the principles of animation originally defined by Disney that are now considered an industry standard (Thomas and Johnston, 1995). Many of these principles are elements of movement that are not found in real life and thus, motion capture alone cannot be used to acquire these movements. The motivation behind our research is to look at how to fill the gap that separates realistic motion from cartoon animation. We want to simplify the animator's work without completely sacrificing their ability to manipulate the result. Thus, our end goal is to allow people to use motion capture data to output cartoon animation.

To speed up the editing process, we propose an algorithm that changes the animation curves by modifying the local peaks and troughs. We focus on exaggerating angular and positional motion to produce more animated results and explore the ability to change an animation's style. We also present a simple curve-based user interface that allows users to visualize potential changes and apply them to the animation. Finally, we present the results of our usability tests on the algorithm, the interface, and the quality of the resulting animations.

2 RELATED WORKS

The ever-growing demand of 3D animation has led researchers to look for innovative ways to either synthesize or modify already existing 3D human motion. The following section describes work that

Ansara R. and Joslin C.
Adding Cartoon-like Motion to Realistic Animations.
DOI: 10.5220/0006174001370147
In *Proceedings of the 12th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2017)*, pages 137-147
ISBN: 978-989-758-224-0

accomplishes this through three different methodologies: We briefly cover space-time constraints and machine learning and focus on motion warping methods, as it is closest to our methodology. Finally, we provide a synopsis of the state of human computer interaction within this context.

2.1 Spacetime Contraints

Spacetime constraints (Witkin and Kass, 1988) defines motion within optimized constraints in direct relationship to the entirety of the action (and is not constrained per keyframe). In this first research paper on the topic, Witkin and Kass successfully allowed *Luxo Jr.*, the lamp mascot from Pixar, to perform multiple different types of jumps based on these constraints. Similarly, Popovic and Witkin attempt to implement dynamics to spacetime constraints with good visual results (Popović and Witkin, 1999). In extension, Li presents a method that allows an animator to edit an already existing animation while preserving the inherent quality of the original motion (Li, Gleicher, Xu, and Shum, 2003) . Comparable research includes the generation of transitions between animations (Rose, Guenter, Bodenheimer, and Cohen, 1996) with the goal of blending different snippets of realistic humanoid animation seamlessly, the creation of a method for the rapid prototyping of realistic character animation (Liu and Popović, 2002) and others.

As mentioned in several of the papers, the main limitation when it comes to animating with spacetime constraints is the need for free-form animations that fall outside pre-determined constraints. These types of animation are difficult to achieve without substantial user intervention.

2.2 Machine Learning

Machine learning uses concepts found in nature to give computer systems the ability to learn and adapt themselves to different situations. The following section presents papers that use various methods of machine learning to teach computers the different aspects of human motion and use that pre-gathered information to generate new, user-defined animations.

Brand and Hertzmann present Style Machines, a system that permits the stylistic synthesis of animation (Brand and Hertzmann, 2000). The focal point of their research is the development of a statistical model that allows a user to generate an array of different animations through the use of various control knobs. Kuznetsova et al. present an innovative technique where the mesh and the motion are combined, synthesizing both human appearance and human motion (Kuznetsova, Troje, and Rosenhahn, n.d.). Furthermore, in Motion Synthesis from Annotations (Arikan, Forsyth, and O'Brien, 2003), Arikan et al. combine constraint-based animations with data-driven methods to present a framework that uses a motion database to generate user-specified animations.

Other work in motion synthesis and editing through machine learning includes the production of poses through a set of constraints and a probability distribution over the space of possible poses (Grochow and Martin, 2004), the classification of stylistic qualities from motion capture data by training a group of radial basis functions (RBF) neural networks (Etemad and Arya, 2014), the combination of IK solvers and a database of motion to create human manipulation tasks (pick-up, place), (Yamane, Kuffner, and Hodgins, 2004) the use of parametric synthesis and parametric motion graphs to produce realistic character animation (Heck and Gleicher, 2007), and the generation of two-character martial art animations (T. K. T. Kwon, Cho, Park, and Shin, 2008).

The main drawback behind machine learning lies in the need to populate the motion database with many different clips to allow for varied results.

2.3 Motion Warping

Motion warping consists of taking a pre-existing motion curve and editing it to get an array of different derivative animations from one source motion. In the following section, we focus on warping systems whose direct goal is to add cartoon effects to realistic motion.

In motion signal processing (Bruderlin and Williams, 1995), the authors introduce techniques from the image and signal processing field for motion editing. They collect all motion parameters from the animation curve of a pre-determined keyframe and convert this into a signal. The large majority of the research this paper discusses focuses on the joint and angle positions of the human hierarchical representation. In their work, Bruderlin and Williams apply signal-processing techniques and modify an animation's trajectory or speed. They apply a multiresolution filtering on a walk animation by increasing the different sub-sections of the different frequencies collected from the original motion signal. Their results show that when they increase the middle frequencies, they obtain a walk-cycle that's both smooth and exaggerated. However, when they choose to only increase the high frequencies, they cause an unnecessary twitching of the character. Although they do not explain the reason, we believe that by increasing those high frequencies, they are dramatizing the effect of the noise in the original data.

Wang et al. presented a method called "The Cartoon Animation Filter" that takes arbitrary input motion and outputs a more animated version of it (Wang and Drucker, 2006). By subtracting a smoothed version of the second derivative of the original curve (of the animation) to the original curve itself, they can add anticipation/follow-through and squash and stretch to a variety of input motions. Their method is the equivalent of applying the inverted Laplacian of a Gaussian filter to the animation curve. In their conclusion, they acknowledge that the resulting animations do not satisfy the expected quality of hand-crafted animations and are more suited for quick, preview-quality animations.

White et al. built on Wang's work by creating a slow in, slow out filter that, when used correctly, can "add energy and spirit" to an animation (White, Loken, and van de Panne, 2006). They accomplish this by identifying eligible keyframes (through Kinematic Centroid Segmentation (Jenkins and Mataric, 2004)) and then applying a time-warp function to obtain the desired result. They demonstrated the effectiveness of their algorithm with a few motion-capture animations, but their paper shows little description of their final result.

Kim et al. (Kim, Choi, Shin, Lee, and Kuijk, 2006) attempted to generate cartoon-like anticipation effects by reversing joint rotation and joint position values. Their results show that their anticipation generation produces more expressive animations and that it's an efficient tool for amateur animators. However, the main drawback of their system is that it heavily relies on user input to define the anticipation and the action stage, a task that proved to be time consuming for some users. Finally, Savoye presented a method that uses Wang et al.'s cartoon filter to allow an animator to modify a pre-existing realistic animation with a single editing point (Savoye, 2011). For example, a user can move an ankle and Savoye's algorithm can be used to estimate the global position of all the other joints in the skeletal hierarchy. He achieves this by minimizing the sum of the squared difference between the original motion capture data and the animator-edited animation data. However, Savoye's results suffer from the same issues as the cartoon animation filter. While the resulting animations do provide adequate cartoon characteristics, the overall quality is lacklustre when compared to keyframed animation and is not recommended as a complete replacement to manual key-framing.

2.4 Novel Animation Systems and User Interaction

While a system's algorithm is a key influence for the quality of the resulting animations, human factor is equally important. The system by Goodwin provides excellent results, but can be difficult to use despite the quality of the animation (Goodwin, 1987); therefore, it can still considered unsuccessful from a practical point of view. In the following section, we will present a short overview of the state of usability considerations in motion synthesis and editing.

Many of the research projects presented above focus on the system functionality and performance without any mention of the human factor involved (J. Kwon and Lee, 2008; Witkin and Popovic, 1995). For example, the cartoon animation filter (Wang and Drucker, 2006) does not describe any method for user interaction. Other work, such as Space-time constraints (Witkin and Kass, 1988), Synthesis of complex dynamic character motion from simple animations (Liu and Popović, 2002), Style Machines (Brand and Hertzmann, 2000), and Synthesis and editing of personalized stylistic human motion (Min, Liu, and Chai, 2010) consider usability issues, but do not formally test for them. There's a considerable amount of interesting interaction systems that have been developed but do not formally conduct user studies. For instance, the "Style Machines" (Brand and Hertzmann, 2000) system uses a series of control knobs to manipulate the difference stylistic changes to the animation and Rose et al.'s method allows users to generate motion transitions through their own custom motion expression language (Rose et al., 1996).

Despite the strong interaction models described in these related works, the main drawback behind these systems is the lack of real user data to support the claims that they're intuitive and easy to use.

3 METHODS

3.1 Motion Capture

To track human motion, we attached reflective markers to participants' bodies by having them wear a tight, black Velcro suit that acts as a membrane and prevents unnecessary reflections from the skin. We recorded all marker data at 120fps using an optical motion capture system from Vicon with six MX40 and four T20 cameras. We used Vicon Blade 1.7.2 for setup, control, and recording; essentially to process the data and output skeletal joint rotations as Euler angles.

3.2 Data Collection

To provide a large variety of input data for our system, we collected several animation clips from a total of 15 participants (8 male, and 7 female) in motion capture sessions that lasted up to 2 hours. We focused on three types of motions: exaggerated joint rotations, unrealistic jumps, and stylistic walk cycles.

Due to the nature of our research goals, our motion captured animations were inspired from a collection of different animated movies. As they are re-enacted by real human beings with real human-proportions, we did not expect the motion capture recordings to be 100% faithful to the animated movie clip. Instead, we used them as a base for interesting recordings that can be both re-enacted by motion capture actors and edited by animators. Because of these constraints, we chose snippets of animations from movies that had a large variety of bipedal characters, such as Toy Story, Rise of the Guardians, Frozen, Despicable Me, The Croods and Hotel Transylvania.

3.3 Plugin Development

We used Autodesk Maya's IDE and Python 2.7 to develop a plugin that allows a user to edit animation data from an imported Vicon Blade skeleton (in FilmBox, FBX, format) directly within Maya.

3.3.1 Exaggeration

Our method is reminiscent of the cartoon animation filter (Wang and Drucker, 2006), which exaggerates the peaks and troughs of a curve to add cartoon-like motion to animation data. However, instead of using a Laplacian Gaussian curve function to dramatize the peaks, we define an exaggerated point (x') as the addition (when greater than the median) or subtraction (when smaller than the median) of a user specified coefficient C. This coefficient is scaled to take the relationship between the original point x, the median (m) and the global maximum or minimum values of the overall animation (M). A visual representation of this method is shown in Figure 1.

$$x' = x \pm (C * ((x - m) / (M - m))) \qquad (1)$$

To reduce the processing power required, we only apply the exaggeration formula above to the pre-determined points and use cubic interpolation to determine the points in between. Figure 2 shows a visual overview of the process.

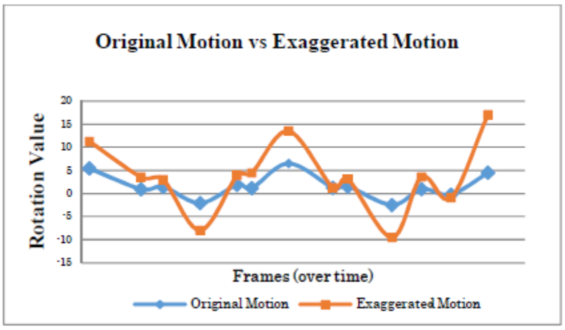

Figure 1: Maximum and Minimum Manipulation.

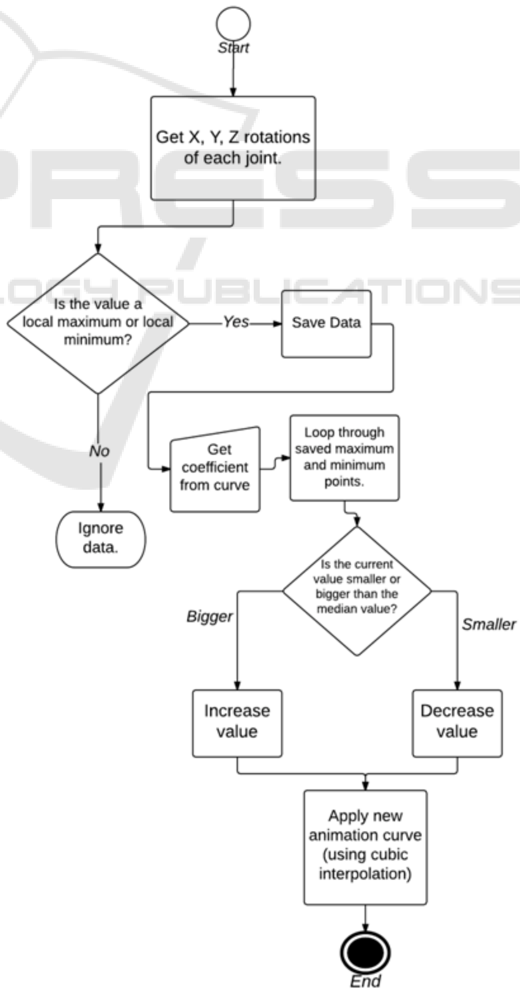

Figure 2: Algorithm Flowchart.

3.3.2 Jump Exaggeration

Our jump trajectory exaggeration feature is an extension to the general exaggeration method. The main difference is that instead of exaggerating the rotation values of all the joints, we specifically target the Y translation value of the root joint and apply our algorithm on that specific axis. This results in jumps with higher amplitude and an added bouncing motion to the landing sequence. Furthermore, to avoid an over-simplification of the jump trajectory, we maintain more than simply the maximum/minimum keyframes.

3.3.3 Exaggerated Feminine Walk

We use the minima and maxima modification algorithm to modify a neutral walk into an exaggerated stylized female walk (exaggerated hip and spine motion) akin to characters such as Jessica Rabbit, Elsa from Frozen, etc. However, instead of increasing the maximum and minimum values uniformly, we target the main features of the walk and modify them accordingly.

In order to determine how to change a neutral walk into a feminine-stylized walk, we collected samples of a neutral walk and of exaggerated female walks. To compare them accurately, we determined the beginning of each walk cycle by looking at the Z position of the foot joint. With the cycles lined up, we examined the difference in amplitudes between our neutral walks versus our exaggerated walks. Using this, we determined the coefficients needed to modify the animation. By adding scaled versions of these values to the stylized walk, we approximate the actual rotation values of an exaggerated feminine walk.

Exaggerating the hip and root joint rotations alone caused a significant amount of feet sliding (more detail in Section 3.3). Consequently, we opted to anchor the feet with IK handles and bring them closer together to give the illusion of an exaggerated hip motion. While this significantly reduces the direct rotational exaggeration, it produces a more feminine walk while removing the undesirable foot sliding artefact.

3.4 Curve Interface

We created a curve-based interface to provide users an easy and straightforward way to edit motion capture clips while showing a clear visual metaphor for the changes that will be applied on the animation. Users manipulate the curve by translating (using the default Maya interface tools) the arrow controls. These shapes are constrained to deformers that move the curve's control points. This effectively morphs the curve depending on how the user manipulates it. This is a simplified version of how animators currently edit animation curves. However, instead of potentially hundreds of control points (x, y, z coordinates, keyframes, etc.), we reduce the interactions to three arrow controls. Figure 3 shows two curves: the first curve, in grey, denotes the default position of the curve. The red curve shows an attempted user edit. The X axis denotes time and the Y axis denotes "amount of change".

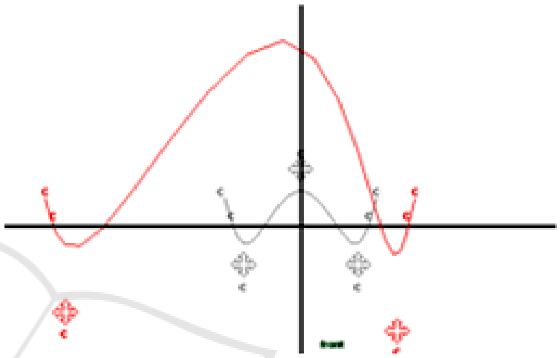

Figure 3: Input curve (larger amplitude, red) in comparison with the default curve position (smaller amplitude, gray).

If users try to apply the default curve to the animation, there is almost no change. In such case, the system finds the local maximum and minimum and re-creates the curves using cubic interpolation, but without changing the peaks or troughs. Conversely, any change in the Y axis will increase the value of local maximums, decrease the value of local minimums and interpolate a new curve based on these new points. The amount of change applied to the local extremes is a scale extracted from the difference between the original curve and the edited curve. Similarly, any change in the X axis applies a uniform time-scale based on the root rotations to emphasize the extreme values of the animation.

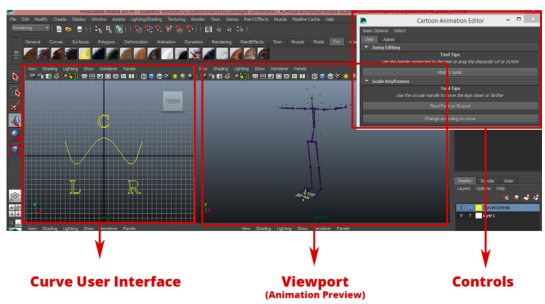

Figure 4: System Overview.

The complete system interface is illustrated in Figure 4. The curve based interface is on the left side of the window, the animation preview window on the centre and the dialog window (on the right side) is mobile and can be moved around to a user's preference.

3.5 Foot Sliding

A ubiquitous problem with editing human motion through joint rotation values is that, because all the joints are organized in a hierarchy, exaggerating the rotation values of the parent joints can cause unexpected results further down. Thus, each exaggeration that is applied creates a successively increasing amount of motion in the end-position of the last joint. While this doesn't affect the upper body negatively, it causes something called foot sliding or foot skating in the lower body, and is generally a problem that still hasn't been completely solved (Kovar, Schreiner, and Gleicher, 2002). As this affects the visual quality of the animations significantly, we implemented a method to correct this issue using inverse kinematic handles to anchor the feet on the floor as necessary. To determine when the foot is bearing weight (and thus, immobile), we find the local minimums of the Y position of the ankle and key-frame the IK handles to those positions. This process prevents the feet sliding effect from happening. While it effectively reduces the overall exaggeration of the lower body, it allows us to avoid awkward animations where character's legs seem to be sliding erratically on the floor.

4 USER STUDIES

4.1 Part A – User Editing

In Part A of our user studies, we asked participants to edit a random selection of ten motion capture animation clips we recorded in our data collection phase. This includes one practice run and three animations per category. All outputted animations were saved along with the curve used to edit them, as well as a reference to their original clip. Our goals were as follows:

- Determine the overall usability of our animation editing system. Will people understand how to edit the animations? Will they grasp the dimensionality of the curve control? Will they find it easy or frustrating?
- Evaluate the visual results produced by the users and verify whether the curve input illustrates the changes in motion effectively. How do they feel about the original animation in comparison with the edited animation? Do they understand the motion change? Which do they prefer? Do they see the relationship between the edited animation and inputted curve?

We recorded user feedback by asking a series of questions related to the animation quality and interactions with the system. The participants filled out a demographics questionnaire before the study began, an animation data sheet after each edited clip, and a final questionnaire at the end of the study to collect their overall opinion of the session.

4.2 Part B – Evaluation

In this second user study, we had a complete different set of participants watch fifteen animations that were created by participants in part A (three of each category) and collected their opinions through Likert-scale questions as well short and long form questions.

Similar to our previous user study, our goals were as follows:

- Compare and evaluate the animation quality by users who were not involved in the creation process and have no emotional attachment to the final product.
- Evaluate the user input as well as the relation of the curve to the animation change.

The participants filled out a demographics questionnaire before the study began, an animation data sheet after each clip they watched, and a final questionnaire at the end of the study to collect their overall opinion of the session.

5 RESULTS

5.1 Algorithm Results

The following chapter shows examples of how our algorithm changes all three animation categories. A quantitative analysis of the participant user experience and rating of the animations is detailed in section 5.2.

5.1.1 Joint Rotation Exaggeration

Our method of exaggerating animation curves is as discussed in Section 3.2 This method is effective at intensifying the maximum and minimum rotational values of the character's joints (namely the root, spine, shoulders, arms, hips, ankles) without over-dramatizing smaller motions. During Part A of our user study, participants exaggerated the overall rotation of a total of 48 dances, walks or runs selected in a randomized order.

5.1.2 Jump Trajectory Exaggeration

As can be seen in Figure 5, our method of jump trajectory modification effectively increases the jump height and the anticipatory motion in an aesthetically pleasing fashion. We asked participants to exaggerate the jump trajectory of three clips, chosen randomly from 48 different animation sources.

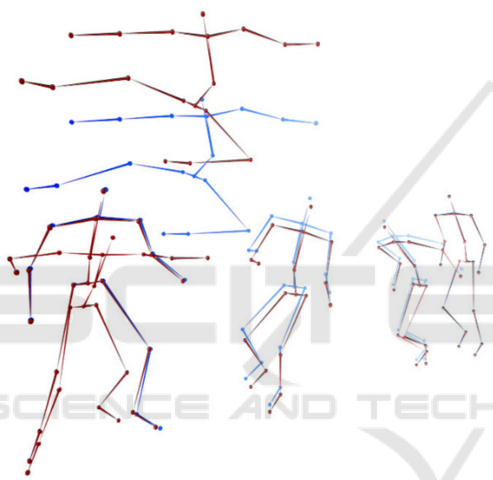

Figure 5: Original (blue) and Edited (red) Jumps.

5.1.3 Neutral Walk to Stylized Walk (Feminine)

Our final application of our algorithm is the stylistic change between a neutral, gender-ambiguous walk to a stylized cartoon female walk. Figure 6 shows an example of an animation resulting from the algorithm as applied to a neutral walk. The edited animation has more dramatic spine movements and reduced arm/shoulder movements when compared to the original animation. The hip motion is constrained because of the foot slide issue, as described in Section 3, which led us to anchor the character's feet. This, in turn, restricted the rotation of the root and hip joints. To create an exaggerated hip motion, we chose to bring the feet closer together. The results of our user study (Section 5.2) show that this gave participants the illusion of more feminine walk.

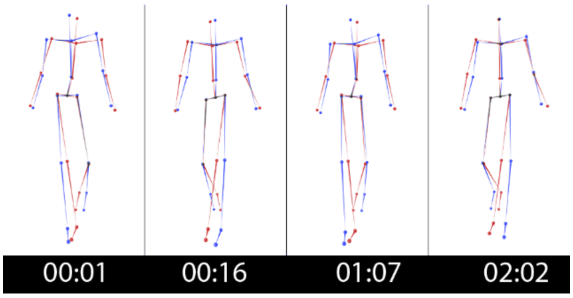

Figure 6: Neutral (blue) and Stylized Walk (red).

5.1.4 Animation Time Shifts

To emphasize the extreme motions of the animations, we allowed users to slow-down the animations elements close to the local minimum and maximums. As previously mentioned, this is achieved by manipulating the X axis in our curve interface. To maintain the homogeneity of the algorithm, we use the root joint as the reference joint to determine when to slowdown the animation. We chose this specific joint for its relative position (at the character's centre of gravity) and for its global influence.

5.2 User Study Results

5.2.1 Part A – Animation Editing with the Curve Interface

In this section, we present the results of the questionnaire administered during Part A of our user studies. We also provide an in-depth statistical analysis of the questionnaire responses. In total, we conducted sixteen one-hour user studies with 16 participants (14 male, 2 female). Each individual edited nine animations (three walks, three jumps, and three miscellaneous clips, chosen at random). The resulting edited animations were used in Part B of the user study.

Participants included undergraduate and graduate students, as well as a few staff. A large majority of our users were between 18 and 34 years old, with only 4 participants over age 35. A majority of participants, 56%, had not done any animation before participating in our study. The remaining participants had experimented with animation in undergraduate courses or through self-learning of various animation methods, such as cell-shading animation, motion capture with Microsoft's Kinect or video editing. Twelve of our sixteen participants (75%) rated their familiarity with Autodesk Maya as very low. In our post-study questionnaire, we evaluated the overall satisfaction level of participants in terms of functionality, interactivity and learning curve of our animation editing system. The results shown in Figure 7 demonstrate that over 80% of participants had a high or very high satisfaction level in all three of our usability criteria.

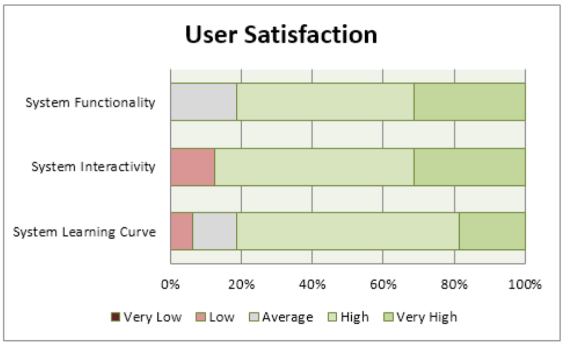

Figure 7: User Satisfaction with the Functionality, Interactivity and Learning Curve of our animation editor.

To further evaluate the usability of our system, we also compared the overall difficulty of our four interaction tasks with the difficulty of three of Autodesk Maya's basic functions:

- Transformations, such as object translation, rotation, and scaling (Move the character)
- Camera manipulations, such as pan, skew, yaw (Move the camera),
- Animation controls, such as play animation, stop animation, change start/end keyframe (Manipulate the timeline)

To do this, we compared participants' average rating of our interaction tasks with their average rating of the Autodesk Maya tasks. Both the average ratings of our interaction tasks, $D(16) = 0.197$, $p = .097$, and the average ratings of the Autodesk Maya tasks, $D(16) = 0.313$, $p = .000$, are significantly different from a normal distribution, as shown by the K-S test. Thus, we compare the two averages using the non-parametric Wilcoxon test. 11 participants rated our interaction tasks as more difficult than the Autodesk Maya tasks, 4 participants rated the Autodesk Maya tasks as more difficult, and 1 participant found them to be equally difficult. However, no statistical difference was found between the average difficulty rating of the basic Autodesk Maya tasks and our animation editing tasks, $Z = -1.936$, $p = .053$. This suggests that our animation tasks are equal in difficulty to basic tasks in Autodesk Maya. However, 75% of our users had no previous experience using Autodesk Maya and yet the average difficulty rating for our interaction tasks was 1.17 (0 -4 scale), which is quite low. This leads us to believe that our curve interface is simple and amateur-friendly, especially compared to the more traditional animation methods discussed in our introduction.

From our observations during the user study, participants found the jump editing to be the most intuitive task, as the arc of the jump visually resembled the amplitude change in the user input. The time-shift task proved slightly more difficult because our implementation didn't allow users fine control over what parts of the animation were slowed down.

Figure 8 shows overall participant satisfaction with the animation quality and with our curve-based interface. 81% (13/16) of participants agreed to some extent that the animations they created were visually pleasing, 75% agreed that the edited versions were more interesting than the original animation and 56% thought the curve was a good visual illustration of the changes done to the animation. It is also interesting to note that none of the participants disagreed with the statement "Overall, the resulting animations I created were more interesting than the original motion". This suggests that our system provides both high functionality and high usability.

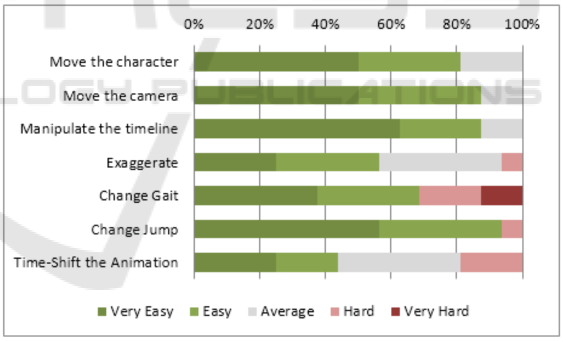

Figure 8: User Satisfaction with the System Functionality, Interactivity and Learning Curve.

5.2.2 Part B – Animation Review

In Part B of our user study, we conducted one-hour sessions with a set of 15 new participants (9 male and 6 female). None of the participants from Part A participated in Part B of the user study. In each session, the participant watched fifteen animations (five jumps, five exaggerations, five walks) and completed a questionnaire, giving us a total of 225 data points. These animations were randomized from the pool of animations generated during Part A of our user study. Of these participants, 50% had experience with basic animation, mostly in Adobe Flash. Only one participant had any motion capture editing experience.

As in Part A of the study, participants in Part B were asked to evaluate how well the edited curve input illustrated the changes made to the animation. Part B participants agreed to some extent that the curve input was a good illustration of changes made to the animation in 63% of the exaggerated clips, 66% of feminized walks and 95% of the jumps. These results lead us to conclude that using a simple curve to illustrate animation changes was a success, particularly when it comes to our jump animations. The jump trajectory exaggeration matches directly with the amplitude increase, likely leading to a strong affordance between the curve interface and the jump animation.

Despite these results, user preferences for the edited and original animations are less promising. Participants indicated that they preferred only 37% of the edited exaggerated animations over the corresponding original clips, 34% of the feminine-stylized walks, and 59% of the edited jumps. All three animation types had lower preferences for the edited animations in this part of the study than in Part A. We explored the reasons for these results by examining the answers to the short and long answer questions from the Part B questionnaire. Participants who liked the animations explained that they found the edited animations "funny", "entertaining", and "unattainable through real motion". Some comments focused on how they preferred animations with less extreme animation edits. Conversely, participants who disliked the edited animations stated that they were "over exaggerated", "unpolished and jerky" or "unrealistic". These results are in line with previous findings: too much exaggeration can cause unappealing results (Reitsma and Pollard, 2003). This over exaggeration is mostly the result of purposeful user input. The lack of realism was the direct object of the interface and the study overall. We believe that perhaps the unrealistic motion would be more appealing to users if the character in the animation was an identifiable cartoon, rather than a bare skeleton; however, this must be left for future work to determine.

5.2.3 Evaluating Curve Input

These results led us to examine the correlation between the participants' preferences for the original vs edited animations with their agreement that sensible curve input was used to change the animations.

By the results of the K-S test, the user opinion of the curve input, regardless of animation type or original vs edited animation, was found to be significantly different from a normal distribution, $D(225) = 0.232$, $p = .000$. Thus the non-parametric Spearman's rank correlation coefficient can be used to correlate this data. 15 participants rated a total of 225 animations, and a significant correlation was found between their agreement that the curve input was sensible and whether they preferred the original over the edited animation, $r_S = .336$, $p = .000$. This correlation indicates that users (in part B) who preferred the original animation were more likely to disapprove of the curve input used by a previous participant (in part A) to edit that animation.

When examining the short and long answer results, we determined a few cases where the animation algorithm didn't work as intended. Most participants described the result of the feminized walk algorithm as feminine, female, or sassy, however a subset found it to be more reminiscent of a drunken person walking, or someone trying to walk on a tight-rope. These impressions can be explained by an over-exaggeration of the spine joint, which makes the character appear unstable. This was then exacerbated by the over-tightening of the feet position described previously. For the two other categories, most criticisms came from frame-skips in the jump animations, and the aggressive use of the time-shift function. Despite this, considering the skill level of participants in Part A of the study, the average preference results are acceptable (43%). Furthermore, we successfully show that the curve interface is a good interaction method that's intuitive and easy to use for amateur users.

6 CONCLUSION

6.1 Findings

In this paper, we presented an animation editing algorithm coupled with a new curve-based interface with the goal of adding cartoon-like qualities to realistic motion. The animation algorithm was based around the idea of interpolating between modified local minimum and maximum values. Our curve interface provides a 2D metaphor to the animation modification process.

The algorithm proved efficient at reducing the realism of the motion across all three animation types, and users found the curve interface easy to use and understand.

6.2 Limitations

Despite the success, our work has a few limitations: Our algorithm does not function very well at extreme data points (large slowdowns, large exaggerations) and can produce very unappealing results when the coefficients are too large (>300% change). In terms of the animation time-shift, our method only allows for the systematic slow-down of the randomly selected clips. To allow for more functionality and a better metaphor between our user interface (the curve) and the results, the ability to speed-up the animation is lacking.

For various reasons discussed in Section 5, our edited animations were not always preferred to the original animations. These results can be explained by a combination of: extreme coefficients applied through the algorithm, user inexperience (at animation) and a lack of context (human skeleton instead of a cartoon character, blank setting instead of a cartoon environment). Furthermore, our user studies, particularly for Part A, pose a few issues. The question “Which animation do you prefer” is vague and subjective and thus gives us scattered results. We suggest changing this to “Which animation is more suitable for cartoon movies?” or another similar question that connects more appropriately to our study goals. In terms of the participant pool, a better balance between genders (a minimal ratio of 40% - 60%) would reduce bias, particularly when it comes to the visual appeal of the exaggerated feminine walk. Finally, to better tie in with our system goals, the participants should have been 3D animators, or at least have had some experience with current cartoon animation methods.

6.3 Future Work

As future work, we’d like to point the research topics in this direction:

Real-time editing: To make the editing process more streamlined, we suggest the implementation of the curve editing system in real- time. This would allow users to make more “on the fly” editing changes and fine tune the results.

Less restrictive foot constraints: Our foot constraints reduced the amount of exaggeration in joints below the hip. While this was necessary to maintain an appropriate level of animation quality, we suggest exploring ways to couple the feet planting process with the exaggeration algorithm to allow interesting modifications of the lower body animations.

Use of cartoon-like character models and settings when editing motion: The usage of humanoid skeletons with realistic proportions poses a few cognitive issues, as certain cartoon motions can look awkward when applied to a realistic human skeleton. This might not be because the animation itself is inherently bad, but rather because it looked out of place. We suggest further research to skin a cartoon-like character to the skeleton to further explore this issue.

ACKNOWLEDGEMENTS

We’d like to thank the School of Information Technology at Carleton University for providing access to their motion capture studio, as well as all the participants who took the time to help us in our data-collection phases and in our user-studies.

This project was funded by The Interactive and Multi-Modal Experience Research Syndicate (IMMERSe).

REFERENCES

Arikan, O., Forsyth, D. A., and O'Brien, J. F. 2003, July. Motion synthesis from annotations. In ACM Transactions on Graphics (TOG) (Vol. 22, No. 3, pp. 402-408). ACM.

Brand, M., and Hertzmann, A. 2000, July. Style machines. In Proceedings of SIGGRAPH, (pp. 183-192). ACM

Bruderlin, A., and Williams, L. 1995, September. Motion signal processing. In Proceedings of SIGGRAPH (pp. 97-104). ACM.

Etemad, S. A., and Arya, A. 2014. Classification and translation of style and affect in human motion using RBF neural networks. Neurocomputing, 129, 585-595.

Goodwin, N. C. 1987. Functionality and usability. Communications of the ACM, 30(3), 229-233.

Grochow, K., Martin, S. L., Hertzmann, A., and Popović, Z. 2004, August. Style-based inverse kinematics. In ACM transactions on graphics (TOG) (Vol. 23, No. 3, pp. 522-531). ACM.

Heck, R., and Gleicher, M. 2007, April. Parametric motion graphs. In Proceedings of the 2007 symposium on Interactive 3D graphics and games (pp. 129-136). ACM.

Jenkins, O. C., and Matarić, M. J. 2004. Performance-derived behavior vocabularies: Data-driven acquisition of skills from motion. International Journal of Humanoid Robotics, 1(02), 237-288.

Kim, J. H., Choi, J. J., Shin, H. J., and Lee, I. K. 2006. Anticipation effect generation for character animation. In Advances in Computer Graphics (pp. 639-646). Springer Berlin Heidelberg.

Kovar, L., Schreiner, J., and Gleicher, M. 2002, July. Footskate cleanup for motion capture editing. In

Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on Computer animation (pp. 97-104). ACM.

Kuznetsova, A., Troje, N. F., and Rosenhahn, B. 2013. A Statistical Model for Coupled Human Shape and Motion Synthesis. In GRAPP/IVAPP (pp. 227-236).

Kwon, J. Y., and Lee, I. K. 2008, September. Exaggerating Character Motions Using Sub□Joint Hierarchy. In Computer Graphics Forum (Vol. 27, No. 6, pp. 1677-1686). Blackwell Publishing Ltd.

Kwon, T., Cho, Y. S., Park, S. I., and Shin, S. Y. 2008. Two-character motion analysis and synthesis. IEEE Transactions on Visualization and Computer Graphics, 14(3), 707-720.

Li, Y., Gleicher, M., Xu, Y. Q., and Shum, H. Y. 2003, July. Stylizing motion with drawings. In Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation (pp. 309-319). Eurographics Association.

Liu, C. K., and Popović, Z. 2002. Synthesis of complex dynamic character motion from simple animations. ACM Transactions on Graphics (TOG), 21(3), 408-416.

Min, J., Liu, H., and Chai, J. 2010, February. Synthesis and editing of personalized stylistic human motion. In Proceedings of the 2010 ACM SIGGRAPH symposium on Interactive 3D Graphics and Games (pp. 39-46). ACM.

Popović, Z., and Witkin, A. 1999, July. Physically based motion transformation. In Proceedings of the 26th annual conference on Computer graphics and interactive techniques (pp. 11-20). ACM.

Reitsma, P. S., and Pollard, N. S. 2003. Perceptual metrics for character animation: sensitivity to errors in ballistic motion. ACM Transactions on Graphics (TOG), 22(3), 537-542.

Rose, C., Guenter, B., Bodenheimer, B., and Cohen, M. F. 1996, August. Efficient generation of motion transitions using spacetime constraints. In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques (pp. 147-154). ACM.

Savoye, Y. 2011, December. Stretchable cartoon editing for skeletal captured animations. In SIGGRAPH Asia 2011 Sketches (p. 5). ACM.

Thomas, F., and Johnston, O. 1995. The illusion of life: Disney animation (pp. 306-312). New York: Hyperion.

Wang, J., Drucker, S. M., Agrawala, M., and Cohen, M. F. 2006, July. The cartoon animation filter. In ACM Transactions on Graphics (TOG) (Vol. 25, No. 3, pp. 1169-1173). ACM.

White, D., Loken, K., and van de Panne, M. 2006, July. Slow in and slow out cartoon animation filter. In ACM SIGGRAPH 2006 Research posters (p. 3). ACM.

Witkin, A., and Kass, M. 1988. Spacetime constraints. ACM Siggraph Computer Graphics, 22(4), 159-168.

Witkin, A., and Popovic, Z. 1995, September. Motion warping. In Proceedings of the 22nd annual conference on Computer graphics and interactive techniques (pp. 105-108). ACM.

Yamane, K., Kuffner, J. J., and Hodgins, J. K. 2004, August. Synthesizing animations of human manipulation tasks. In ACM Transactions on Graphics (TOG) (Vol. 23, No. 3, pp. 532-539). ACM.