Procedural Animation of Human Interaction using Inverse Kinematics and Fuzzy Logic

Gaetan Deglorie[1], Koen Samyn[2], Peter Lambert[1], Rik Van de Walle[1] and Sofie Van Hoecke[1]

[1]*ELIS Department, Multimedia Lab, Ghent University-iMinds, Gaston Crommenlaan 8, Bus 201, B-9050 Gent, Belgium*

[2]*DAE, University College West Flanders, Ghent University Association, Botenkopersstraat 2, 8500 Kortrijk, Belgium*

Keywords: Procedural Animation, Multi-character Interaction, Inverse Kinematics, Fuzzy Logic.

Abstract: Nowadays, animation of human interaction is predominantly modelled statically. Animations are adapted manually to each set of participating characters to ensure visual fidelity. To automate this process, we propose a novel procedural animation technique where character interactions are modelled dynamically with fuzzy logic and compare our technique to conventional inverse kinematics. The 'handshake' interaction is used as an example interaction, in order to illustrate how basic animation rules are defined, while leaving room for parametrization of character specific properties. Our results show that, although inverse kinematics delivers higher precision in positioning than fuzzy logic, they are dependent on paths describing the motion of the final element in the kinematic chain. Fuzzy logic, on the other hand, is independent of such motion paths and solves towards the target location locally. The presented handshake model using fuzzy logic can serve as a basis for future models for virtual-human interaction.

1 INTRODUCTION

Video games, movies and simulations often contain scenes with character interaction. To create animations for these interactions, a large fraction of production resources is spent on manually producing or recording these animations through motion capture. Although the resulting animations do not adapt at runtime without extra processing steps, this property is highly desirable in dynamic virtual environments. One approach towards adaptive animation is procedural animation, where animations are generated algorithmically. Two important problems exist for this approach: (1) synchronization of separate animation sequences of participating characters, and (2) real-time synthesis in order to generate animations for a specific context in a real-time environment.

In this paper, we compare two procedural animation techniques for animating human interactions:

1. inverse kinematics, where limbs are retargeted based on joint parameters obtained from kinematic equations (Johansen, 2009)
2. fuzzy logic controllers, where the joint parameters are locally controlled for each joint (Samyn et al., 2012).

Both procedural animation approaches are compared using a handshake test case.

The remainder of this paper is as follows. After an overview of related work in Section 2, the methodology of our approach is presented in Section 3. Section 4 describes a case study of real-life handshakes, as well as the concluded rules for defining the virtual model. Next, in Section 5, the underlying framework facilitating animations of human interactions is presented. In Section 6 the model of the handshake is described in detail. The results are presented in Section 7. Finally, the conclusions and future work are summarised in Section 8.

2 RELATED WORK

In this section, the state-of-the-art on procedural animation and multi-character animation is reviewed.

Procedural Animation

Multiple variants of procedural animation exist; we list the ones most relevant to our work.

First, *animation retargeting* adapts an existing animation to a new context or situation. E.g., the walk cycle of a character on a flat surface can be adapted to slanted surfaces. This technique can be implemented using inverse kinematics (Johansen, 2009).

Deglorie G., Samyn K., Lambert P., Van de Walle R. and Van Hoecke S..
Procedural Animation of Human Interaction using Inverse Kinematics and Fuzzy Logic.
DOI: 10.5220/0005310603400347
In *Proceedings of the 10th International Conference on Computer Graphics Theory and Applications* (GRAPP-2015), pages 340-347
ISBN: 978-989-758-087-1

Second, *physics-based character animation* utilizes real-time physics simulation to animate the limbs of articulated skeletons. This way, high visual fidelity is achieved, but the animations lack the high level of control that is required by traditional animation artists (Geijtenbeek and Pronost, 2012).

Finally, a third approach procedurally animates the walk cycle of a character using *fuzzy logic controllers* (Samyn et al., 2012). In this paper we extend this approach to be usable for multi-character animation.

Multi-character Animation
The creation of animation for multi-character interactions has been approached in literature in a number of ways.

First, a combined physics-based and data-driven approach creates multi-character motions from short single-character sequences (Liu et al., 2006) and defines the motion synthesis as a spacetime optimisation problem. This method animates the approach of two hands towards each other, by iteratively drawing them together until convergence is reached. However, this solution is not suitable for real-time synthesis.

Secondly, Kwon models competitive interaction (i.e. Taekwondo) as a dynamic Bayesian network (Kwon et al., 2008). It synchronizes the animation of multiple characters by timewarping the appropriate sequences of motion capture data. Whereas Kwon uses multi-character motion capture, Schum uses motion capture data of a single person (Shum et al., 2012). Here character interactions are simulated by expanding a *game tree* of possible states and evaluating these states towards the future to select the appropriate animation sequence. Both approaches are unable to adapt their respective animations to different character configurations.

A novel *Laplacian motion editing method* lets artists manipulate synchronized multi-character animation through the use of synchronized motion paths (Kim et al., 2009). These motion paths describe the position of the character in space and time, while constraint-based displacement is used to constrain specified limbs to a target location (e.g., characters' hands constraints when carrying a chair).

to achieve synchronization, adaptation and real-time synthesis.

3 METHODOLOGY

The state-of-the-art approaches to animate multiple characters mostly address issues concerning synchronization of individual animations and adaptation through motion re-targeting. They however do not deal with real-time synthesis, which will become a necessity with the ever increasing variation of characters in virtual environments. To solve this shortcoming, a novel procedural animation technique based on fuzzy logic is presented to achieve synchronization, adaptation and real-time synthesis.

We compare this fuzzy logic based techniques to create multi-character animation using parametrization of procedural techniques with inverse kinematics (IK). Whereas conventional IK is used to retarget existing animations (Johansen, 2009), we use it to retarget kinematic chains (i.e. the right arms of the characters) to keyframed motion paths in order to overcome the need for captured data. The IK controller calculates the joint parameters of the kinematic chain for each frame, based on the current path position. Contrary to IK, the fuzzy logic approach solves towards the target location locally (i.e. considering joints independently) in incremental time steps and allows the use of both motion paths and static targets.

Our method supports *synchronization* and *adaptation*. To create believable animations of human interactions, the individual animations of each character have to be *synchronized*. By constraining the end effectors (i.e. final joint in a kinematic chain) of multiple characters to the same motion path, they perform synchronized animations. The animation adapts to all partaking characters by parametrizing the motion paths and/or targets, specifically to their position, dimensions and personality (i.e. dominance).

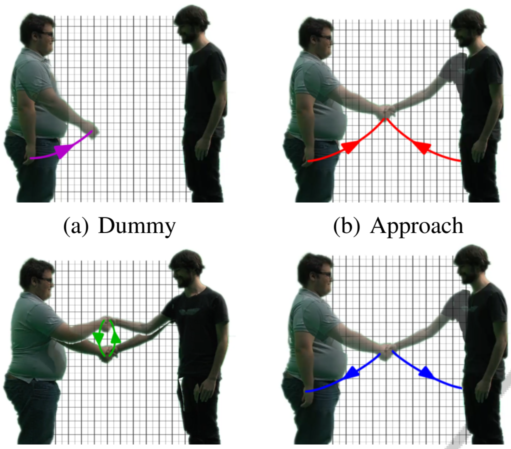
4 CASE STUDY

Our animation model is based on a two-part case study that captured handshake gestures between pairs of people. We identify four phases within the handshake interaction: *dummy phase* (the starting subject invites the other subject to shake hands), *approach phase* (both hands move towards each other to initiate the grip), *shake phase* (the hands move up and down) and *retreat phase* (the subjects retract their hands back towards their body). The different phases are displayed in Figure 1.

In the case study, we focus on the shake phase, examining the flow and dimensions of the shaking motion and the orientation of the hands during the hand grip in particular.

4.1 Shaking Motion of the Hands

The test used five test subjects (named A-B-C-D-E), paired into three groups: (1) male A - male B, (2) fe-



(a) Dummy (b) Approach

(c) Shake (d) Retreat

Figure 1: The subsequent phases of the handshake.

male C - female D, and (3) male E - female D. The handshake was performed five times by each group to mitigate the effect of possible outliers in the captured data. While performing the handshake experiments, the subjects stood on pre-placed markers so that the data on the arm movements could be compared. The shake phase of each handshake was analysed using a top and a side view (see Figure 2). We refer to the starting point of the shake phase as the “point of contact”.

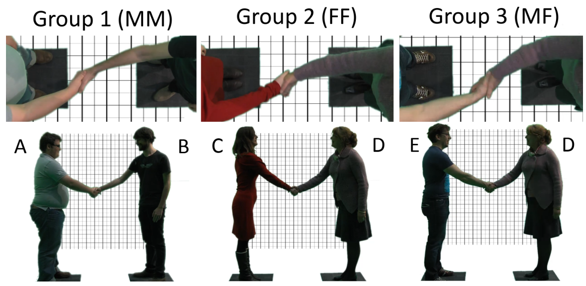


Figure 2: Stills of the 3 groups during part one of the case study.

Considering the overall flow, we observed both single-shake (group 1) and multi-shake (groups 2 and 3) interaction. In the latter two groups, the multiple shakes can be separated as one large primary shake and multiple smaller secondary shakes. These secondary shakes can be accredited to subject D. We only focus here on the primary shake.

The video footage used for this part of the case study can be found at http://vimeo.com/107897073.

4.1.1 Top View Analysis

During analysis of the top view, we observed two characteristics. First, during the shake phase the joined hands stay in line with both (right) elbows, i.e. the handshake grip and elbows create an initial plane perpendicular to the floor, in which the entire shake phase occurs. Second, on average the point of contact is centred in between the subjects. Slight deviations can occur as a result of difference in arm length (group male-male), subject position or dominance between subjects. We infer dominance of a person from the rule that a dominant person keeps the shake away from his/her personal space (Lewis, 2012).

Table 1: Results of angle analysis for the 3 groups.

	Group 1 male-male		Group 2 female-female		Group 3 male-female	
	Avg.	Std. Dev.	Avg.	Std. Dev.	Avg.	Std. Dev.
EF	3.2	1.2	0.5	3.0	3.9	2.2
SF	92.5	2.9	77.2	11.5	86.2	3.5
SE	89.4	2.2	76.7	14.2	82.3	3.9

4.1.2 Side View Analysis

The vector of travel for both hands was also studied, i.e. the direction and distance travelled during the shake phase, referred to as “shake direction” and “shake distance” respectively. To infer both, we used the video frames corresponding to the lowest and highest position during the shake phase. By overlaying these frames, the figure can be simplified as a quadrilateral (as represented in Figure 3). We observed that in some cases the elbow moves between the lowest and highest position. In that case, we used the position of the lowest point to construct the quadrilateral. Next, we determined the orientations

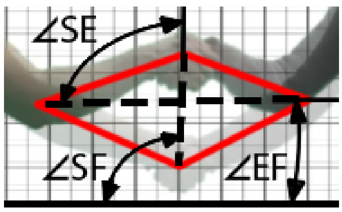


Figure 3: Side view handshake of group 3 with highlights.

of the diagonals (indicated by the dashed lines). The horizontal diagonal is referred to as the *elbow-line* and the vertical diagonal as the *shake-line*. We observe three angles (illustrated in Figure 3): (1) between the elbow-line and the floor (EF), (2) between the shake-line and the floor (SF), and (3) between the shake-line and the elbow-line (SE). As can be seen in Table 1, EF is always close to zero. Deviations can be explained by either the difference in height (e.g. for group 1) or a result of the personalities of the subjects. The results of group 1 show that SE should be a right angle. In the cases of group 2 and 3, we attribute the deviations from this value to an implied difference in mannerism. SF is the superposition of angles SE and EF. For the direction of travel during the shake phase, we use SE and constrain it to be a right angle (i.e. shake-line perpendicular the elbow-line).

The shake distance is calculated as a proportion of the elbow-line length. No large deviations are observed for this proportion. On average across all cases, the final ratio of shake distance to distance between elbows is *35 percent*. This value is used as an input for the generation of the vector of travel during the shake phase, i.e. the observed effect in the resulting animation. More detail is given in Section 7.

4.2 Orientation of the Hands

Next, we define the specific orientation and positional offset required to model a handshake grip. In our model both hands are assumed equal in size. To

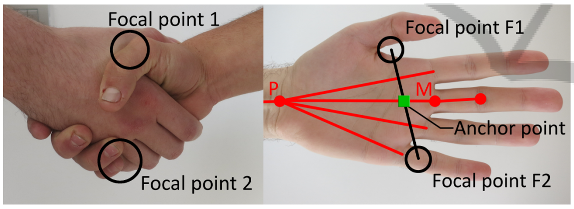


Figure 4: Handshake grip with focal points (left) and single hand analysis (right).

model the handshake grip, two focal points are identified. As can be seen on Figure 4, these focal points on each hand coincide with the matching focal points of the other hand and are aligned vertically during the grip. The intersection of the line between the focal points and the middle finger palm bone (i.e. third metacarpal bone) serves as the center of rotation, indicated by the "anchor point" square. The angle of rotation is defined as the required rotation to place line F1-F2 in a vertical position. Next, a small offset reduces the implicit clipping behaviour of both hands. This offset is defined as half the thickness of the palm, in the direction of the normal vector leaving the palm.

4.3 Complete Handshake Model

Based on the case study of real life handshakes, a set of rules is created for defining a virtual handshake model. These rules consist of:

1. the initial point of contact of both hands located between the subjects (weighted according to dominance);
2. the vector of travel of the shake phase (35 percent of the distance between the elbows; in the direction perpendicular to the elbow-line); and
3. the orientation and offset of the hands during the shake phase (based on the dimensions of the hand).

5 FRAMEWORK

Based on the handshake case study, a framework for the creation and playback of interaction animations is designed.

We personify virtual humans through personality traits represented by key-value pairs (trait name - trait value). Traits such as dominance can be specified and compared in order to adapt animations accordingly. We parametrize the handshake animation model through the personality trait of dominance.

A handshake is an example of an interaction where one subject starts the interaction and the other subject accepts (or refuses) this initiation. The animation framework should therefore support interactions with an asynchronous start. Accordingly, an arbiter component is introduced (see Figure 5) in order to support this behaviour while doubling as a skeletal pose sharing service. For reusability, the arbiter is designed to be animation- and controller type agnostic.

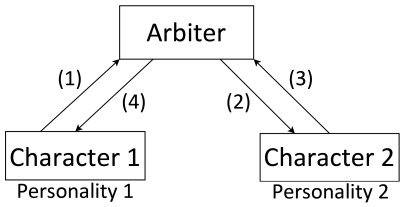


Figure 5: Interaction work flow with arbiter.

The arbiter handles the asynchronous start and shares information between both parties (more specifically sharing the current skeletal pose of the other subject). If the interaction is accepted, the arbiter passes a reference to the data of each subject to the other. The architecture of the arbiter is completely independent of the handshake animation and controller type, allowing the arbiter to be used in future for other similar types of animation (e.g. high fives).

6 HANDSHAKE ANIMATION

We now show how to implement the handshake interaction in our framework. We describe the motion of the hands using the four handshake phases defined in Section 4: dummy, approach, shake and retreat. The initiating character starts with the dummy phase, here the hand moves a short distance in front of the character. Next, both characters go into the approach phase, moving the hands from a neutral position (or current position for the initiating subject) to the point of contact. The shake phase then, calculates the vector of travel and moves the hands upward and downward along this vector. During this phase, the motion of both hands is synchronized using a shared motion path. Furthermore, a geometric handle is used to enforce rotation and offset of both hands. Finally, the retreat phase makes the hands return to a neutral position or an ongoing overarching animation.

The transitions between phases are modelled as a finite state machine (FSM) as shown in Figure 6. State transitions are handled individually with the exception of the transition from A1/A2 to S1/S2 where both parties need to agree (i.e. synchronize) in order to switch states.

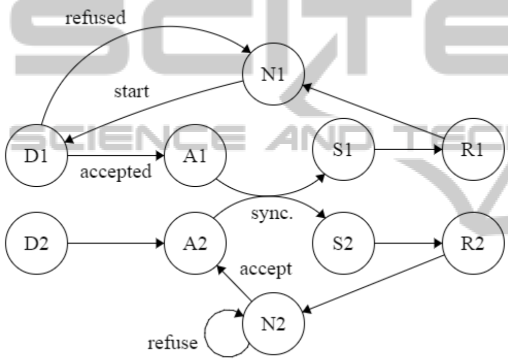


Figure 6: Handshake phases modelled as a finite state machine with states: N1 (neutral state of initiating subject), N2 (neutral state of other subject), D (dummy phase), A (approach phase), S (shake phase) and R (retreate phase).

For every phase, each character initializes a *partial animation* and adapts it according to the position, dimensions and personality of the character. As partial animations are initialized right before the start of each phase, the animation can adapt to parametrical or environmental changes on a per-phase granularity.

In the following subsections, we discuss the use of IK and fuzzy logic animation controllers in the framework.

6.1 Inverse Kinematics

Our IK approach uses keyframed motion paths to guide the character animation, i.e. the end effector of the kinematic chain moves along this path. The motion paths are represented by B-splines, facilitating personalisation of the animation. The path is constructed as the shortest distance between two points: the current location of the hand and the target location for the end of that phase (e.g. the point of contact defines the end of the approach phase).

IK has many known implementations (Aristidou and Lasenby, 2009), two of the more popular being Jacobian Transpose IK and Cyclic Coordinate Descent IK (CCD). Both alternatives are compared in Section 7.

6.2 Fuzzy Logic

In the fuzzy logic approach, a fuzzy logic controller is attached to each individual joint of the kinematic chain. The input of the fuzzy logic controllers is *angular error* α_i. Angular error is determined locally for each joint, and is defined by the rotation required to move the end effector in line with the target position (see Figure 7). We let the joint angles approach their target state at a speed that scales with the angular error of their respective joint. Conceptually, both the input and output have five *membership functions* (Zadeh and Kacprzyk, 1992). Angular error has memberships: Far(left), Near(left), Zero, Near(right) and Far(right); speed has memberships: Fast(left), Slow(left), Zero, Slow(right) and Fast(right).

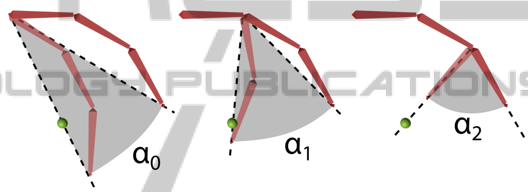


Figure 7: Angular error α_i of each joint in the kinematic chain.

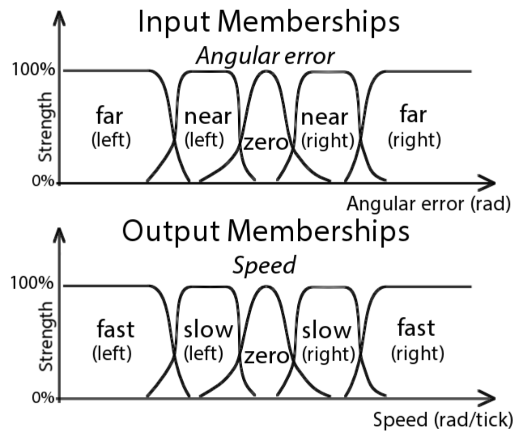


Figure 8: Input and output memberships for the fuzzy logic controllers.

We use the following rules for each controller:

```
IF angularError IS FarL THEN speed IS FastR
IF angularError IS CloseL THEN speed IS SlowR
IF angularError IS Zero THEN speed IS Zero
IF angularError IS CloseR THEN speed IS SlowL
IF angularError IS FarR THEN speed IS FastL
```

From this, the controller returns a fuzzy output for the speed of the joint. The output is defuzzified using a *centroid defuzzifier* to obtain the matching scalar value for the speed. By keeping the output values for speed low, all joints rotate in small increments over time.

Hence, it is possible to animate using only a static target position, instead of requiring motion paths as was the case with IK.

The animator has no exact control over the path taken by the end effector towards such static target positions. If however control is wanted, the animator can choose to use motion paths instead.

By adding rotational information to the target position, we can extend our approach to support axial rotation of the individual bones, which in turn can reduce artefacts during transitions between phases.

7 RESULTS

The handshake animation can be configured in two important ways, on one hand the choice of animation controllers in both characters, and on the other hand the influence of character properties such as position, scale and personality. Video footage of all configurations can be found on http://vimeo.com/107897074.

7.1 Animation Controller Configuration

We evaluated four test configurations for both animation controller *independence* and animation controller performance.

Table 2: Four configurations of animation controllers.

#	Controller 1	Controller 2
1	Jacobian Transpose	Jacobian Transpose
2	CCD	CCD
3	Fuzzy logic	Fuzzy logic
4	CCD	Fuzzy logic

For each test configuration, we evaluated the positional precision of the end effector or right hand of each character during the shake phase. Three metrics were calculated as a measure of error: (1) the distance between both end effectors, (2) the distance between the hand of character 1 and the target, and (3) the distance between the hand of character 2 and the target.

The resulting values are presented in Figure 9. Note that distance is expressed in engine units, corresponding to centimetres in the real world.

The summarized results for all graphs are displayed in Table 3. For errors, we calculated both the mean and the standard deviation. Execution times are also shown as a measure of performance (measured as number of CPU ticks per frame on an Intel(R) Core(TM) i7-2600 3.40GHz).

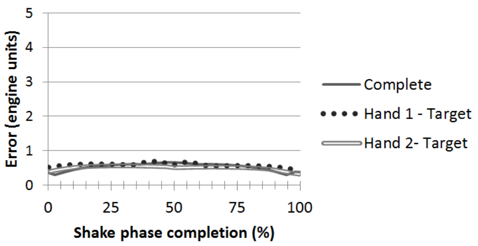
Configuration 1 achieves an overall high accuracy with only a small decrease halfway the animation, as shown in Figure 9(a). However, this high level of precision comes at the cost of a comparatively higher execution time (see Table 3).

Configuration 2 has an overall lower precision than Jacobian Transpose with a comparatively better precision halfway the shake (see Figure 9(b)). Additionally, the video footage shows that the hands do not maintain their relative alignment. This is due to the iterative rotation of joints in CCD, which makes the final bone in the chain align with the target. This is particularly visible at the end of the approach phase and start of the retreat phase.

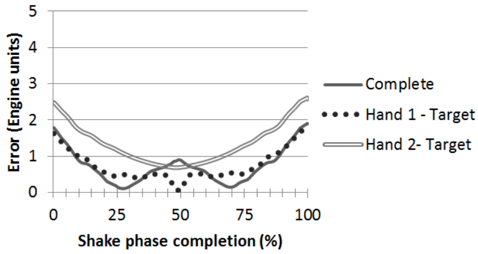
Configuration 3 has both subjects using fuzzy logic controllers. As can be seen in Figure 9(c) and Table 3, the precision of this configuration has an overall lower precision than with both IK methods. The path of the gradient shows that fuzzy logic has reactive behaviour. The error increases because it is unable to keep up with the animation target. At the peak of the shake phase, the target switches direction and moves towards the delayed end-effector, decreasing the error severely. After passing the end effector, the error increases again towards the end of the shake phase. The complete error shows similar behaviour, which can be accredited to a small difference in movement speed of both hands.

Configuration 4 shows the hybrid controller approach. The performance, shown in Figure 9(d), is not surprisingly a superposition of the separate error rates. It shows how our underlying interaction framework operates independently of specific animation controller types. To our knowledge, this is not possible in the current state-of-the-art.

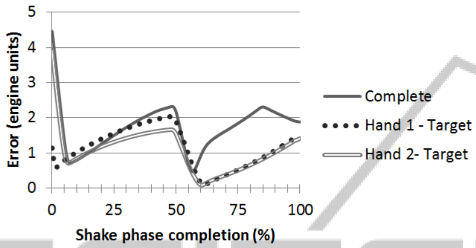
As can be seen in Table 3, Jacobian Transpose IK (configuration 1) performs with the lowest error rate in all cases, but at the expense of having the lowest computational performance. CCD IK (configuration 2) has a slightly higher error rate, but has the best performance of all. Fuzzy logic (configuration 3) offers a tradeoff with an error rate that is higher than CCD and a performance in between CCD and Jacobian Transpose. The resulting precision of the hybrid configuration (configuration 4) is an average between



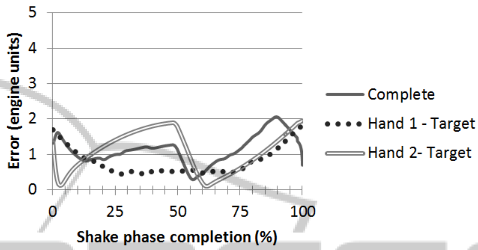
(a) Jacobian Tranpose



(b) CCD



(c) Fuzzy logic



(d) Hybrid (Hand 1:IK / Hand 2: Fuzzy logic)

Figure 9: Error measurement for each configuration.

Table 3: Summary of animation controller influence.

Configuration	Jacob. Transpose(1)		CCD(2)		Fuzzy logic(3)		Hybrid(4)	
Error	*Avg.*	*Std.Dev.*	*Avg.*	*Std.Dev.*	*Avg.*	*Std.Dev.*	*Avg.*	*Std.Dev.*
Complete	0.53	0.10	0.72	0.46	1.70	0.61	1.14	0.40
Hand 1	0.58	0.05	0.73	0.40	1.13	0.56	0.78	0.37
Hand 2	0.50	0.06	1.35	0.55	1.06	0.57	1.09	0.57
Performance (ticks/fr.)	1250		300		800		N/A	

fuzzy logic and CCD. We believe that the fuzzy logic approach can reach similar precision as CCD in the future. The fuzzy logic error curve shows delayed following behaviour, which can be reduced by tweaking the controller response speed.

7.2 Character Configuration

We now discuss the adaptability of the animation to changes in position, dimensions and personality. The effect of changes in character position is presented in Figure 10. As can be seen, the orientation and position of the handshake adapts automatically according to the model described in Section 4. However, as we only animate from the clavicle to the hand, the current model has a limited range of valid character positions.

Figure 11 shows the effect of scale difference. The complete handshake automatically adapts to the scale, except for the handshake grip. Again, the range of possible scale differences is limited, as our human model cannot bend over to shake hands with very small characters.

Finally, in Figure 12 the effect of difference in personality is displayed. Both characters have a personality trait “dominance” which influences the point of contact. The handshake grip shifts towards the less dominant character (the left character in this figure).

Figure 10: Varying configurations of character position.

Figure 11: Variation of character scale.

Figure 12: Variation of character personality.

8 CONCLUSION AND FUTURE WORK

In this paper, we presented a handshake animation model as a use case for general multi-character interactions. The model *synchronizes* specific parts of the animation in order to achieve believable interactions, whilst *adapting* to changes in position, dimensions and personality of the participating characters. Compared to the current state-of-the-art, our approach synchronizes multiple single character animations and is suitable for real-time animation synthesis. We compared inverse kinematics with fuzzy logic. Whereas inverse kinematics achieves a higher precision than fuzzy logic, it however relies completely on motion paths. Contrary, fuzzy logic can operate without them. When using static targets with fuzzy logic, the animator cannot control the path taken by the end effector towards these targets. In that case motion paths can still be used to enforce a certain path.

Both presented approaches use motion paths to facilitate *synchronization* in an animation controller type agnostic manner. This enables us to create hybrid interactions where both characters use different animation controller types.

Future work can encompass, on one hand, creating alternative animations to handshakes and, on the other hand, increasing the detail of this interaction model. For fuzzy logic, future work can be the use of machine learning to tweak the individual controllers, the addition of axial rotation control to the end effector, and handling of non-constant speed values throughout the phases. Furthermore, more in depth case studies should be done in order to achieve animations with higher (visual) fidelity.

ACKNOWLEDGEMENTS

The authors would like to thank DigitalRune GmbH for providing a non-commercial license of their library. This work was partly funded by the Friendly Attac project (http://www.friendlyattac.be/), a project of the Institute for the Promotion of Innovation through Science and Technology in Flanders (IWT).

REFERENCES

Aristidou, A. and Lasenby, J. (2009). Inverse kinematics: a review of existing techniques and introduction of a new fast iterative solver.

Geijtenbeek, T. and Pronost, N. (2012). Interactive character animation using simulated physics: A state-of-the-art review. In *Computer Graphics Forum*, volume 31, pages 2492–2515. Wiley Online Library.

Johansen, R. S. (2009). Automated semi-procedural animation for character locomotion. Master's thesis, Aarhus University, Department of Information and Media Studies.

Kim, M., Hyun, K., Kim, J., and Lee, J. (2009). Synchronized multi-character motion editing. In *ACM Transactions on Graphics (TOG)*, volume 28, page 79. ACM.

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

Lewis, H. (2012). *Body language: A guide for professionals*. SAGE Publications India.

Liu, C. K., Hertzmann, A., and Popović, Z. (2006). Composition of complex optimal multi-character motions. In *Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 215–222. Eurographics Association.

Samyn, K., Van Hoecke, S., Pieters, B., Hollemeersch, C., Demeulemeester, A., and van de Walle, R. (2012). Real-time animation of human characters with fuzzy controllers. In *Proceedings of The 2012 Computer Graphics International Conference*. Computer Graphics Society.

Shum, H. P. H., Komura, T., and Yamazaki, S. (2012). Simulating multiple character interactions with collaborative and adversarial goals. *Visualization and Computer Graphics, IEEE Transactions on*, 18(5):741–752.

Zadeh, L. and Kacprzyk, J. (1992). *Fuzzy logic for the management of uncertainty*. Wiley professional computing. Wiley.