Learning Two-Person Interaction Models for Responsive Synthetic Humanoids

David Vogt*, Heni Ben Amor†, Erik Berger*, Bernhard Jung*

*Institute for Informatics
TU Bergakademie Freiberg
Bernhard-von-Cotta-Straße 2 09599 Freiberg
{david.vogt|erik.berger|jung}@informatik.tu-freiberg.de
http://humanoids.tu-freiberg.de

†FB Informatik
TU Darmstadt
Hochschulstraße 10 64289 Darmstadt
amor@ias.tu-darmstadt.de
www: http://www.ias.tu-darmstadt.de/Member/HeniBenAmor

Abstract

Imitation learning is a promising approach for generating life-like behaviors of virtual humans and humanoid robots. So far, however, imitation learning has been mostly restricted to single agent settings where observed motions are adapted to new environment conditions but not to the dynamic behavior of interaction partners. In this paper, we introduce a new imitation learning approach that is based on the simultaneous motion capture of two human interaction partners. From the observed interactions, low-dimensional motion models are extracted and a mapping between these motion models is learned. This interaction model allows the real-time generation of agent behaviors that are responsive to the body movements of an interaction partner. The interaction model can be applied both to the animation of virtual characters as well as to the behavior generation for humanoid robots..

First presented at Virtuelle und Erweiterte Realität - 9. Workshop der GI-Fachgruppe VR/AR 2012, extended and revised for JVRB

Keywords: Imitation learning, motor learning, motion adaptation, interaction learning, virtual characters, humanoid robots

1 Introduction

Generating natural behavior for synthetic humanoids such as virtual characters and humanoid robots in interactive settings is a difficult task. Many degrees of freedom have to be controlled and coordinated so as to realize smooth movements and convincing reactions to the environment and interaction partners. A common approach for the generation of human-like behaviors is based on motion capture data. Movements recorded from human demonstrators are replayed and possibly slightly altered to fit the current situation.

So far, however, approaches based on real-time adaptation of motion capture data have been mostly restricted to settings in which the behavior of only one agent is recorded during the data acquisition phase. As a result, the mutual dependencies inherent to the interaction between two agents cannot be represented and, thus, reproduced. During a live interaction, a synthetic humanoid may not be able to appropriately respond to the behavior of a human interaction partner.

Moreover, due to recent developments in cheap motion capture technologies, e.g. the Kinect camera, there is an increasing need for algorithms that can generate responses to perceived body movements of a human interaction partner. E.g. in gaming, a virtual character should recognize the behavior of the human player and respond with an appropriate reaction.

In this paper, we present a novel approach for realizing responsive synthetic humanoids, that can learn to react to the body movements of a human interaction partner. The approach extends traditional imitation learning [BCDS08] to settings involving two persons. Using motion capture technology, the movements of a pair of persons are first recorded and then processed using machine learning algorithms. The resulting interaction model encodes which postures of the passive interaction partner have been obtained depending on postures of the active interaction partner. Once a model is learned, it can be used by a synthetic humanoid (passive interaction partner) to engage in a similar interaction with a human counter part. The presented interaction learning approach addresses not only the question of what to imitate, but also when to imitate (c.f. [DN02]).

This paper is organized as follows: In Section 2 we discuss related work. In Section 3 we introduce two-person interaction models and describe how to learn them from the simultaneous motion capture of two interaction partners. Section 4 shows examples how two-person interaction models can be applied to the behavior generation of virtual characters. In Section 5, we discuss two alternative machine learning algorithms for training our two-person interaction models. In section 6 we discuss the applicability of two-person interaction models to humanoid robots briefly before we finally conclude in Section 7.

2 Related Work

Motion capture has been widely used for animating virtual characters. However, adapting the acquired motion data to new situations is a difficult task that is usually performed offline. In contrast, interactive settings as considered in our approach require the real-time adaptation of motion capture data. Some of the previous approaches for this are discussed in the following.

In [YL10], for example, feedback controllers are constructed that can be used to adapt motion capture data sequences to physical pertubations and changes in the virtual environment. The approach formulates motion tracking as an optimal control problem whereby optimization methods derive parameters of a controller for real-time generation of full-body motions.

Multon and colleagues [MKHK09] present a framework for animating virtual characters, where motion capture data is adapted in real-time based on a morphology-independent representation of motion and space-time constraints.

Ishigaki et al. [IWZL09] introduced a control mechanism that utilizes example motions as well as live user movements to drive a kinematic controller. Here, a physics model generates a character's reactive motions.

Other work aims to adapt motion capture data through the inclusion of inverse kinematics solvers, e.g. [KHL05, MM05].

Generating natural looking motions is also a focal point in the field of humanoid robotics. Researchers have extended the concept of motion capture by introducing imitation learning techniques, which can learn a compact representation of the observed behavior [BCDS08]. Once a motion representation is learned, it can be used to synthesize new movements similar to the shown behavior while at the same time adapting them to the current environmental conditions. Imitation learning, thus, aims to combine the advantages of model-driven approaches, such as adaptability to unknown execution contexts, with benefits of data-driven approaches, such as synthesis of more natural-looking motions.

In an motion generation approach presented by Calinon et al. [CDS+10] Gaussian Mixture Regression models and Hidden Markov models (HMMs) are used to learn new gestures from multiple human demonstrations. In doing so time-independent models are built to reproduce the dynamics of observed movements.

In [YT08] the authors utilize recurrent neural networks to learn abstract task representations, e.g. pick and place. During offline training a robots' arms are guided to desired positions and the teachers' demonstrations are captured.

In recent years, various attempts have been undertaken for using machine learning in human-robot-interaction scenarios. In [WDA+12], an extension of the Gaussian Process Dynamics Model was used to infer the intention of a human player during a table-tennis game. Through the analysis of the human player's movement, a robot player was able to

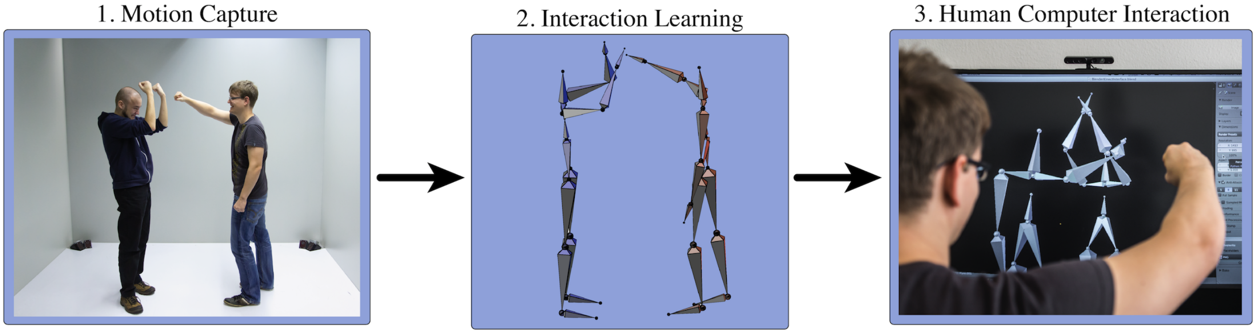

Figure 1: Use of two-person interaction models: Two humans' live-motions are captured, e.g. using the Kinect depth camera, and projected to a low-dimensional posture space. This space is used to learn a model of the shown interaction which is then utilized for motion generation for a synthetic humanoid in real-time.

determine the position to which the ball will be returned. This predictive ability allowed the robot to initiate its movements even before the human hit the ball. In [IAM$^+$12], Gaussian mixture models were used to adapt the timing of a humanoid robot to that of a human partner in close-contact interaction scenarios. The parameters of the interaction model were updated using binary evaluation information obtained from the human. While the approach allowed for human-in-the-loop learning and adaptation, it did not include any imitation of observed interactions. In a similar vein, the work in [LN10] showed how a robot can be actively involved in learning how to interact with a human partner. The robot performed a previously learned motion pattern and observed the partner's reaction to it. Learning was realized by recognizing the observed reaction and by encoding the action-reaction patterns in a HMM. The HMM was then used to synthesize similar interactions.

However, in all of the above approaches, the synthetic humanoid's motion is generated from motion demonstrations by only one actor. In contrast, the approach presented below draws on the simultaneous motion capture of two actors that demonstrate example human-human interactions.

3 Learning Two-Person Interaction Models

In this section, a novel interaction learning method is introduced that allows virtual characters as well as humanoid robots to responsively react to the on-going movements of a human interaction partner (see Figure 1). At the core of our approach is an interaction model which is learned from example interactions of two human actors.

The interaction model is built in three steps: (1) Interactions between two persons are recorded via motion capture; (2) the dimensionality of the recorded motions is reduced; and, (3) a mapping between the two low-dimensional motions is learned.

3.1 Motion capture of two-person interactions

In the first step, the movements of two people interacting with each other are captured. In general, the presented interaction learning approach is independent of particular motion capture systems. For development and testing, we used the Kinect sensor to record joint angles.

The precision of the calculated joint angles depends on the sampling rate. In general, low sampling rates lead to small datasets lacking accuracy whereas high sampling rates result in larger datasets with increased precision. Also, higher sampling rates may lead to redundant joint angle values for slow motions. For the behaviors used in this paper a sampling rate of 30fps is used.

Recorded motions are very high dimensional as 48 joint angles are captured per actor. A problem that arises when recording such motions is that not all measured variables are important in order to understand the underlying phenomena [EL04]. Hence, dimensionality reduction should be applied to remove the redundant information, producing a more economic representation of the recorded motion. To this end, we use Principal Component Analysis (PCA) for construction of a low-dimensional posture space [Amo10] which will be explained further in the following section.

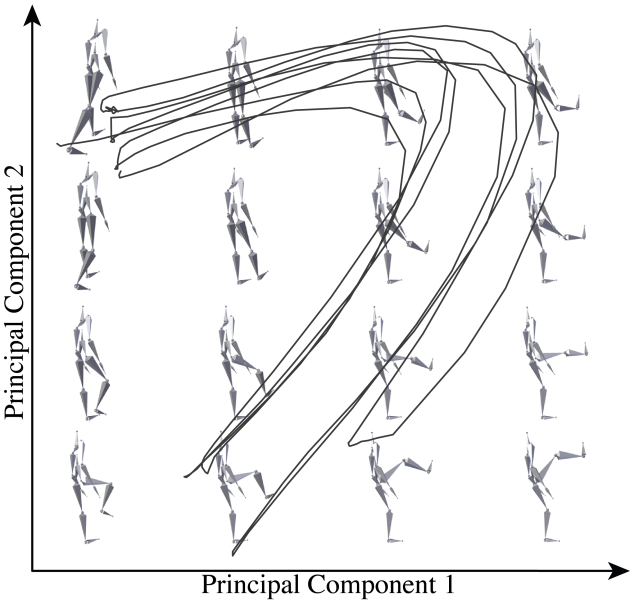

Figure 2: A low-dimensional posture space for a kick behavior reduced to two dimensions with principal component analysis. Every point in this space corresponds to a posture which can be reprojected to its original dimensionality and adopted by a virtual human or humanoid robot.

3.2 Dimensionality Reduction

Principal Component Analysis (PCA) is applied to the high-dimensional motion capture data to yield a low-dimensional posture space [Amo10]. PCA reduces the dimensionality of a dataset based on the covariance matrix of modeled variables. Dimensionality reduction is achieved by finding a small set of orthogonal linear combinations (the principal components) of the original variables depending on the largest variance.

Previous results indicate that for most skills 90% of the original information can be retained with two to four principal components [ABVJ09]. Each point in the posture space corresponds to a pose of the synthetic humanoid as illustrated in figure 2. Accordingly, a trajectory in posture space corresponds to a continuous movement. Hence, new behaviors can be created by generating trajectories in the posture space and projecting these back to the original high-dimensional space of joint angle values [Amo10].

3.3 Two-Person Interaction Models

A two-person interaction model is the combination of two low-dimensional posture spaces with a mapping function from one posture space to the other. It is important that every point in the input posture space can be mapped to the output posture space and not just a subset of points, e.g. points lying on the low-dimensional trajectory of the demonstrated movement. In this way, the interaction model can also map postures that are similar to but not exactly like postures seen during training.

In the following, two machine learning algorithms with suitable generalization capabilities for learning such mapping functions are described.

3.3.1 Feedforward Neural Net

Learning input-output relationships from recorded motion data can be considered as the problem of approximating an unknown function. A Feedforward Neural Net (FNN) is known to be well suited for this task [MC01].

In our experiments, we use a simple FNN consisting of three layers, i.e. an input, a hidden and an output layer. How many neurons the hidden layer consists of and which connectivity value has to be used, depends on the recorded motion data. In general, the net should just have enough neurons to fit the data adequately while providing enough generalization capabilities for complex functions. We use 10 neurons in the hidden layer since we found that this fits the data adequately while retaining generalization capabilities for unseen low dimensional points. More neurons could increase generalization but also increase the risk of overfitting the data.

For the training of a FNN Levenberg-Marquardt backpropagation is utilized and all points from the low-dimensional trajectory of the active interaction partner are used as training data. Overfitting is avoided by using *early stopping*. For that low dimensional data is divided into two subsets: for training and testing.

The training data is used to adapt the nets weights and biases. The test data is utilized to monitor the validation error. This error normally decreases for a number of iterations. When it starts increasing again, this indicates overfitting and the training is stopped.

Generally, not only the current posture but also its history is necessary to determine an appropriate response. As FNNs have no short-term memory, the input to the FNN cannot be limited to the current posture. Instead, the FNN takes as input a sequence of low-dimensional postures, i.e. the current posture and a number of history postures.

The labels (output) of the net are all low-

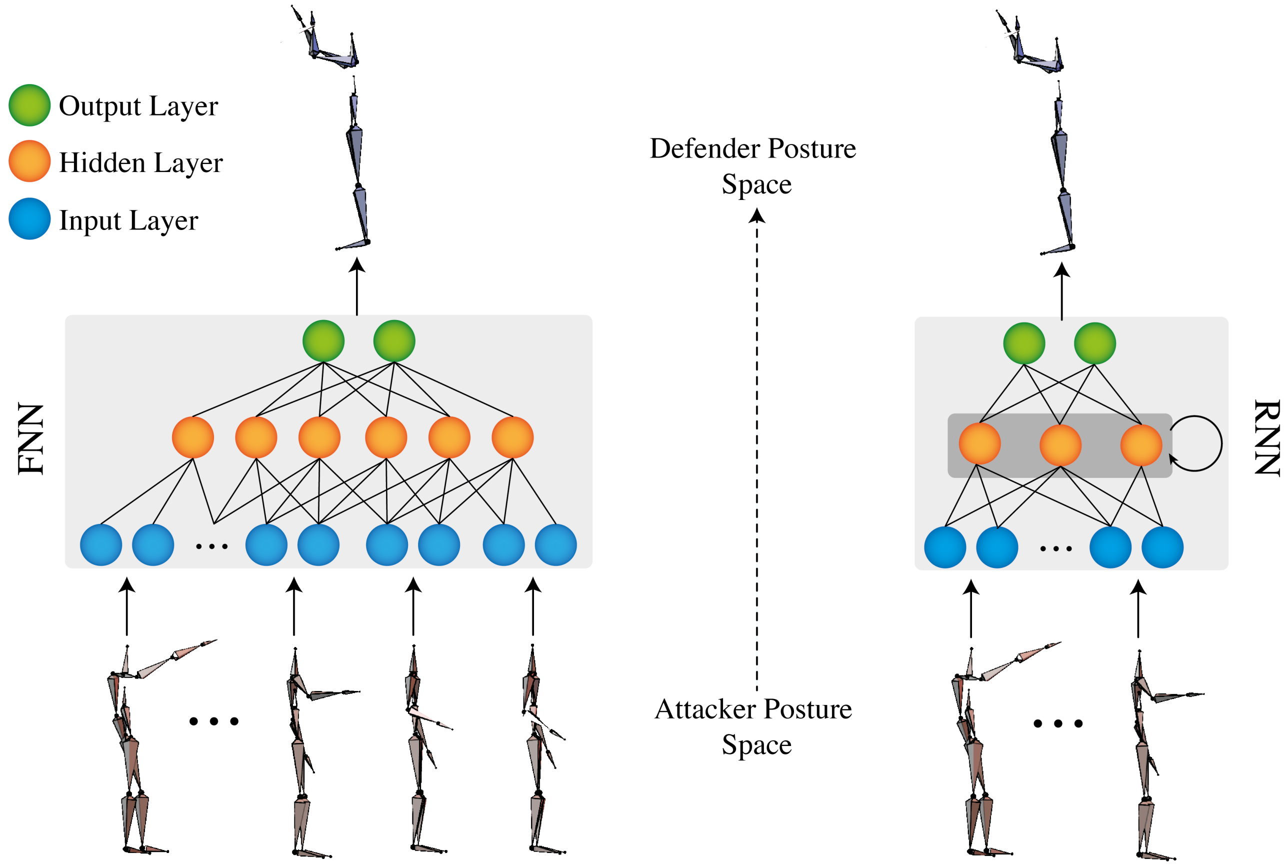

Figure 3: Multiple human postures are mapped on single robot/virtual human posture. Input to both nets (Left: FNN; Right: RNN) are the low-dimensional postures from a sliding window over time, i.e. the current posture with a history of preceeding postures. The output is the low-dimensional posture that represents the robot's/virtual human's response to the currently observed posture.

dimensional points on the low-dimensional trajectory of the second, reactive interaction partner. Figure 3 shows how a sequence of postures is mapped onto a single output posture using an FNN.

3.3.2 Layered Recurrent Neural Network

Strict feedforward neural nets have no short term memory and cannot store history postures. Desired memory effects are only created due to the way how past inputs are represented in the net. Hence, a sliding window has to be used. And, thus, the amount of input neurons increases with each additional pose by the number of dimensions of the low-dimensional posture space.

To circumvent this problem we utilize layered recurrent neural nets (RNN) [Elm90] which have proven to be well suited for modeling various temporal sequences [FS02, YT08, NNT08]. RNNs have a feedback activation in the hidden layer where each hidden neuron feeds back to all hidden neurons. This embodies the desired short term memory. The hidden layer is updated not only with the current external input but also with activations from the previous timestep. This allows the usage of smaller sliding windows and, therefore, less input neurons. In our experiments the amount of neurons in the hidden layer is the same as for feedforward neural nets.

For training, we use Bayesian regulation backpropagation. Training examples are low-dimensional points of the active partner's movements and their mapping onto low-dimensional points on the trajectory of the reactive interaction partner (see figure 3). The trained RNN defines a continuous mapping from the input posture space to the output posture space while preserving the temporal context of the recorded interaction.

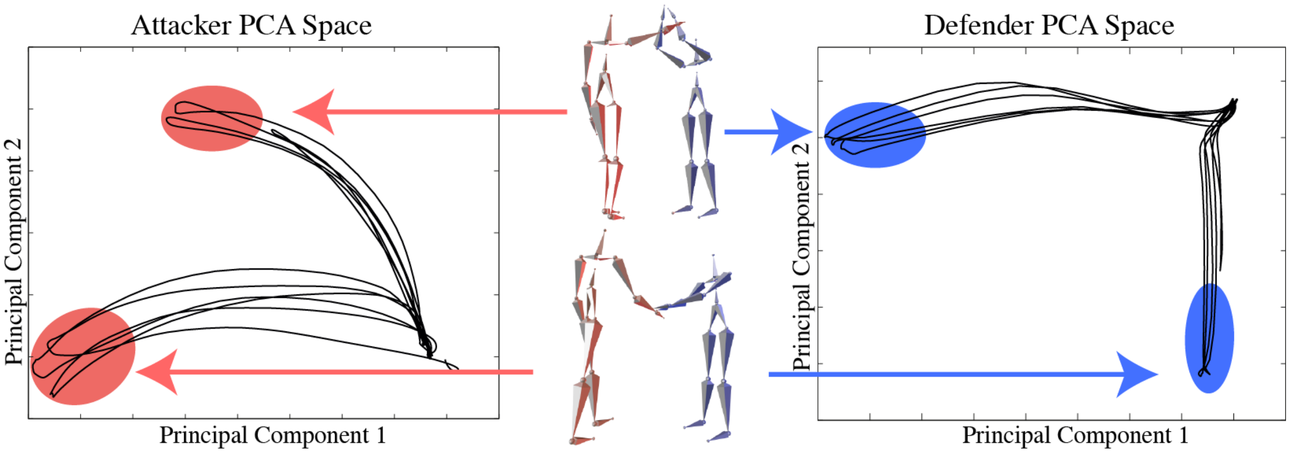

Figure 4: Two different punches and defenses have been used to learn an interaction model. Left: A low punch is defended with both arms pointing downwards. Right: A high punch is defended by pulling both arms up for defense.

4 Examples

In the following, we describe two examples where two persons are recorded while performing martial arts. The first interaction is a combination of punches varying in height with proper defenses. The second scenario consists of different kicks also with suitable defenses. For both behaviors, a virtual human learns to block the attacks with the trained defenses.

The motions of both the attacker and the defender start and end in an upright position with both arms stretched. The Kinect sensor was used as motion capture device in both examples.

4.1 Punching

In the first example a two person interaction model is used to generalize various punches and continuously compute motions for a defending virtual character. For that, two behaviors have been recorded. The first is a sequence of low punches at stomach level where the defender stretches his arms forward to block the attack. The second behavior consists of high punches at face level where the defender had to pull up both arms for a proper defense. In the center of figure 4 both punches with their defense motion are shown.

The training of the mapping algorithms is based on these two recordings. For each behavior, three repetitions of the respective punching style and their defense are captured. Then, the motions are projected into low dimensional space which can be seen in figure 4 for both interaction partners.

All low dimensional points of the behavior trajectories are utilized to train both mapping algorithms. In our experiments a 10 point pose history for FNNs, i.e. a 0.66 second window, has proven to be well suited. For RNNs good learning results have been achieved with only two points in the history.

For live-interaction with a virtual character, the user's current posture is captured at 30fps and projected into the previously created low-dimensional posture space. In combination with previous points a sequence of history poses is used as input for the mapping algorithm which predicts a low-dimensional point in the virtual character's posture space. The resulting point is then projected back to its original dimension and used as target pose for the virtual character.

Figure 5 shows how the learned two person inter-

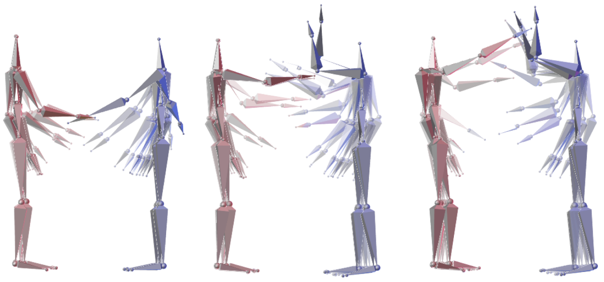

Figure 5: The image shows three punches for a user driven avatar (red skeleton) and the calculated virtual characters defense motion (blue skeleton). The user's motion on the left and right were similar to the ones used for model learning. However, the punch height in the middle has not been trained explicitly. Still, the learned model can calculate plausible motions.

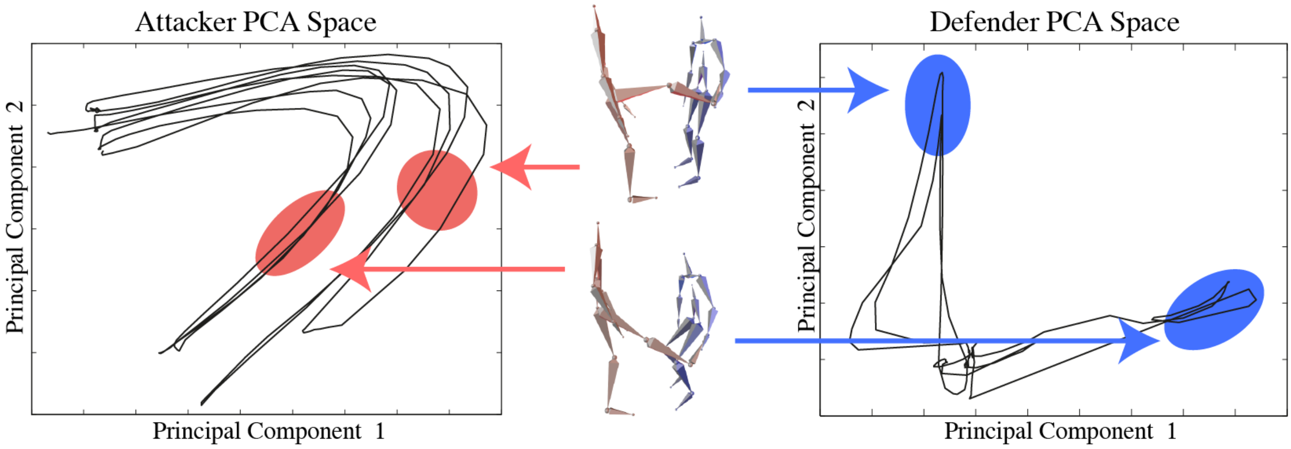

Figure 6: Two different kicks and defenses were used to learn an interaction model. For low kicks the defender needs to crouch down in order to reach the attackers leg.

action model calculates the postures of a virtual character (blue skeleton) depending on postures obtained by a user driven avatar (red skeleton). The left and right of the figure display a newly recorded user motion similar to the ones used to train the mapping algorithm. However, the motion displayed in the center of the figure exhibits a punch height that has not been trained explicitly. Nevertheless, the interaction model can calculate suitable motions for the defending virtual character.

With the learned mappings, the virtual character can defend itself against the trained attacks. A clear differentiation between punching heights has been learned. Even for varying punch heights, the virtual human can still respond with correct defense motions.

4.2 Kicking

In the second example a two-person interaction model is learned from various kicks allowing a virtual character to defend itself with suitable defenses. For that two different kicks have been recorded. The first is a low kick where the defender needed to bend its knees in order to reach the attackers' foot. The second is a higher kick which was defended with an upright position (see figure 6).

For model learning both attack styles were performed three times and combined in a single animation. Since each motion started and ended in an upright position, an enclosed trajectory is created in low-dimensional space, as can be seen in figure 6.

This 4-dimensional posture space contains enough information to reconstruct animations with a negligible error. The neural nets were configured to consist of a hidden layer with 10 neurons. Addtionally, the size of the pose history has been set to 5 for FNNs and 2 for RNNs respectively.

For a live interaction with the character, the user's current posture is once again fed into the mapping algorithm and an appropriate pose for the defending character is predicted. Figure 7 shows a user controlled character repeating various kicks which where similar to the ones used to drive the nets training. Additionally, a kick height (figure 7 center) not present in the recordings was performed. Since, varying kick heights of the attacker result in different low dimensional points, the virtual characters trajectories are located in the range of the previously recorded motion.

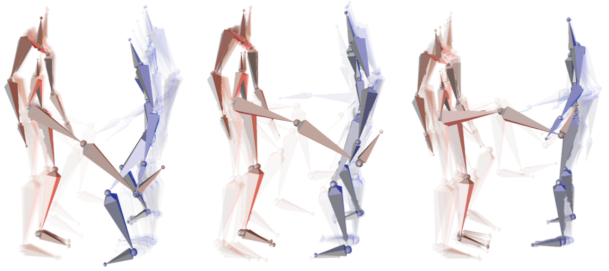

Figure 7: A learned interaction model is used to calculate a virtual character's defend motions (blue character) based on a user controlled avatar (red character). The kick heights on the left and right are similar to the ones obtained during training. However, the kick height in the center was not present in any recording. Nevertheless, the two-person interaction model can generate suitable motions for the defending virtual character.

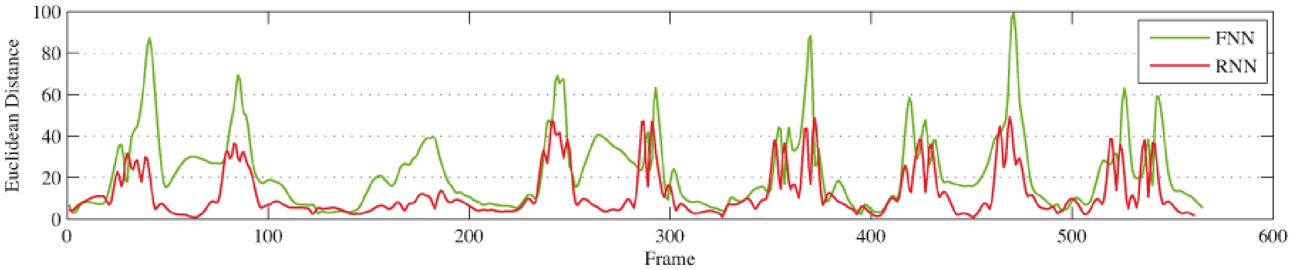

Figure 8: Euclidean distances are used to validate both mapping algorithm's qualities. Here, the graph shows distances for a novel punch data set. As it can be seen RNNs exhibit a smaller overall error.

For all three kick heights the virtual character learned when to defend itself using a crouched defense and when to block in an upright posture. In both cases the arms need to block the attacker's leg at the right time. For similar kicks, yet with different heights, proper defenses are generated by both FNN and RNN.

The interaction model that has been learned during this example allows the control of a virtual character based on demonstrations of only two variants of the behavior. The character is controlled continuously in a low-dimensional space and the model is robust to unseen user postures. Since the executed behaviors are based on human motion data, a life-like appeal of virtual characters can been produced.

5 Comparison of Mapping Algorithms

The interaction models for the examples were learned with both mapping algorithms described above, i.e. FNN and RNN. In order to evaluate the quality of the mappings, the Euclidean distances between desired and trained low-dimensional points are measured, e.g. the training error. For that we recorded another punching example that is similar to the original human motion. A comparison of the resulting error is shown in figure 8 for the punching example.

Overall, we found that both mapping algorithms produce similar results. However, FNNs tend to exhibit weaker generalization capabilities. With varying interaction learning scenarios, differing net sizes were used in order to minimize the training error. Since the required size is hard to determine beforehand a cumbersome, trial and error process is inevitable.

In contrast, RNNs are more tolerant towards untrained inputs and no architecture changes have been required throughout the examples. Furthermore, RNNs maintain a short term memory, which renders posture histories unnecessary. However, we found that by the inclusion of a small number history poses, i.e. two additional postures, smoother responses are generated by the RNN. Figure 9 shows the prediction for training data with various history sizes. As can be seen, the overall response time of the synthetic humanoid increases if additional postures are used to predict the character's pose. We found that 2-point histories, i.e. the original point and two previous points, represent a good compromise between fast response times and high smoothness of the generated movements. This corresponds to the works of Ziemke et al. [Zie96] which came to a similar conclusion. The combination of recurrence and a small sliding window tends to work best.

6 Applicability to Humanoid Robots

Two person interaction models can also be used to control humanoid robots, although an additional processing step to adapt the resulting motion is necessary. It is generally known that human motion cannot be transfered to robots without further optimization and is generally referred to as the correspondence problem [ANDA03]. To overcome this the low-dimensional trajectory of the defender has to be altered to fit the robot's kinematic chain and stability constraints. For that we utilize an inverse kinematics solver.

The two-person interaction model thus maps the original low-dimensional movement of the attacker to the adopted low-dimensional movement of the defender (robot). The learned interaction model is then used to predict robot postures depending on observed user postures. As can be seen in figure 10 a Nao robot successfully mimics the demonstrated defense behavior.

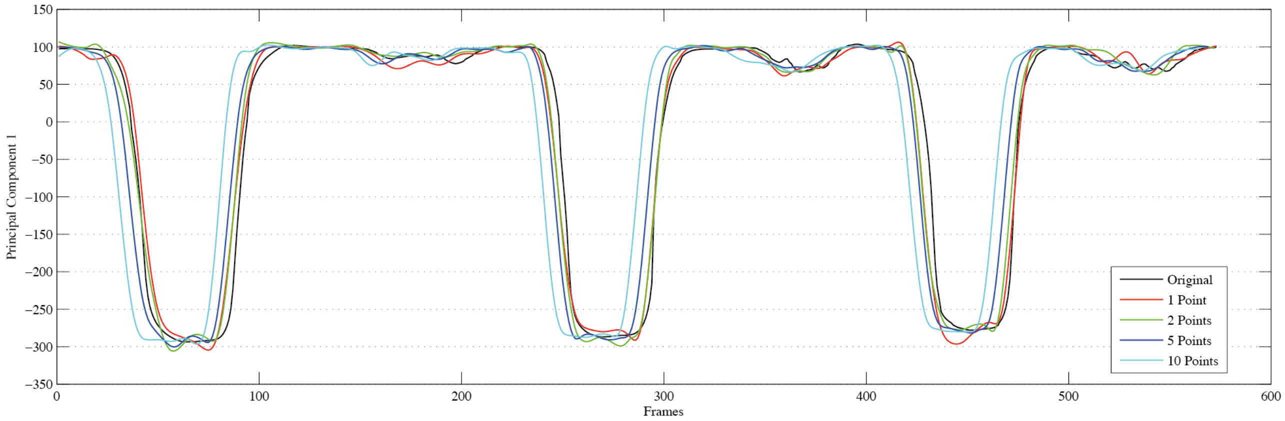

Figure 9: A pose history improves the overall mapping quality. The figure illustrates how various sizes influence smoothness and response time for the punch behavior of a RNN. With increasing history sizes the net's prediction is shifted to the left, leading to larger response times but at the same time smoother motions are generated. We found that three points, i.e. the original point and two previous points in the posture history produced the best results.

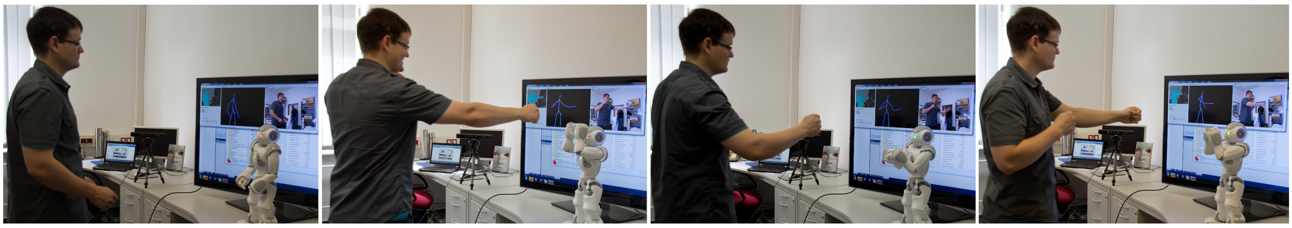

Figure 10: Utilizing a two-person interaction model, a humanoid robot can react to the on-going behavior of a human interaction partner. The robot's motion is controlled by a low-dimensional mapping which preserves temporal coherences of the trained behavior.

7 Conclusion

In this paper we presented a new approach for teaching synthetic humanoids, such as virtual characters and humanoid robots, how to react to human behavior. A recorded example of the interaction between two persons is used to learn an interaction model specifying how to move in a particular situation. Two machine learning algorithms have been implemented to establish a mapping between (low-dimensional) movements of the two persons: Simple feedforward neural networks (FNN) and layered recurrent neural networks (RNN). While the FNN requires a large sliding window of current and recent body postures as input, the RNN can store previous poses in its internal state and thus can produce appropriate responses with a small sliding window of history poses. In our experiments, the RNN performed slightly better than the FNN.

In contrast to previous approaches to interaction modeling found in computer animation and game theory, no explicit segmentation of the seen behavior into separate parts is required. The responses of the synthetic humanoid are calculated continuously, based on learned mapping between the body postures of the observed humans. As a result, the interaction model can generalize to different situations while at the same time producing, smooth continuous movements.

Our method can be used to learn how to control synthetic humanoids from observation of similar situations between two humans. This can help to significantly increase the realism in the interaction between a robot and a human, or a virtual character and a human.

So far, however, our approach is based on joint angle data and, hence, response generation is based on similarities in joint-space. However, in order to more accurately reflect critical spatial relationships between the body parts of the interaction partners, motion synthesis could also be based on optimizations in task-space. By doing so, we could also better account for varying body heights of both interactants.

An interesting addition to the current approach would be a higher-level component for planning and strategic decision making. This could, for example, be realized through a combination of our approach and the approach of Wampler et al. [WAH+10]. We also hope to model multiple possible responses per stimulus. In our current setup each movement of the human can trigger only one possible response of the virtual character. A mixture-of-experts approach [RP06], in which several interaction models are combined, can help to overcome this problem.

References

[ABVJ09] Heni Ben Amor, Erik Berger, David Vogt, and Bernhard Jung, *Kinesthetic bootstrapping: teaching motor skills to humanoid robots through physical interaction*, Proceedings of the 32nd annual German conference on advances in artificial intelligence (Berlin, Heidelberg), KI'09, Springer-Verlag, 2009, pp. 492–499, ISBN 978-3-642-04616-2.

[Amo10] Heni Ben Amor, *Imitation Learning of Motor Skills for Synthetic Humanoids*, Ph.D. thesis, Technische Universität Bergakademie Freiberg, 2010.

[ANDA03] Aris Alissandrakis, Chrystopher L. Nehaniv, Kerstin Dautenhahn, and Hatfield Herts Al Ab, *Solving the Correspondence Problem between Dissimilarly Embodied Robotic Arms Using the ALICE Imitation Mechanism*, Proceedings of the second international symposium on imitation in animals and artifacts, 2003, pp. 79–92.

[BCDS08] Aude Billard, Sylvain Calinon, Ruediger Dillmann, and Stefan Schaal, *Handbook of Robotics*, ch. 59: Robot Programming by Demonstration, pp. 1371–1389, MIT Press, 2008.

[CDS+10] Sylvain Calinon, Florent D'halluin, Eric L. Sauser, Darwin G. Caldwell, and Aude G. Billard, *Evaluation of a probabilistic approach to learn and reproduce gestures by imitation*, Robotics and Automation (ICRA), 2010 IEEE International Conference on, 2010, pp. 2671–2676.

[DN02] Kerstin Dautenhahn and Chrystopher L. Nehaniv (eds.), *Imitation in animals and artifacts*, ch. The agent-based perspective on imitation, MIT Press, Cambridge, MA, USA, 2002.

[EL04] Ahmed Elgammal and Chan-Su Lee, *Inferring 3D Body Pose from Silhouettes Using Activity Manifold Learning*, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (Los Alamitos, CA, USA), vol. 2, IEEE Computer Society, 2004, pp. 681–688, ISBN 0-7695-2158-4.

[Elm90] Jeffrey L. Elman, *Finding structure in time*, Cognitive Science **14** (1990), no. 2, 179–211, ISSN 0364-0213.

[FS02] Eberhart E. Fetz and Larry E. Shupe, *The hand book of brain theory and neural network*, ch. Recurrent network: neurophysiological modeling, pp. 860–863, MIT Press, Cambridge, 2002.

[IAM+12] Shuhei Ikemoto, Heni Ben Amor, Takashi Minato, Bernhard Jung, and Hiroshi Ishiguro, *Physical Human-Robot Interaction: Mutual Learning and Adaptation*, Robotics Automation Magazine, IEEE **19** (2012), no. 4, 24–35, ISSN 1070-9932.

[IWZL09] Satoru Ishigaki, Timothy White, Victor B. Zordan, and C. Karen Liu, *Performance-based control interface for character animation*, ACM SIGGRAPH 2009 papers (New York, NY, USA), ACM, 2009, Article no. 61, ISBN 978-1-60558-726-4.

[KHL05] Taku Komura, Edmond S. L. Ho, and Rynson W. H. Lau, *Animating reactive motion using momentum-based inverse kinematics: Motion Capture and Retrieval*, Comput. Animat. Virtual Worlds **16** (2005), no. 3-4, 213–223, ISSN 1546-4261.

[LN10] Dongheui Lee and Yoshihiko Nakamura, *Mimesis Model from Partial Observations for a Humanoid Robot*, Int. J. Rob.

Res. **29** (2010), no. 1, 60–80, ISSN 0278-3649.

[MC01] Danilo Mandic and Jonathon Chambers, *Recurrent neural networks for prediction: learning algorithms, architectures and stability*, Adaptive and learning systems for signal processing, communications, and control, John Wiley and Sons, Inc. New York, NY, USA, 2001, ISBN 0471495174.

[MKHK09] Franck Multon, Richard Kulpa, Ludovic Hoyet, and Taku Komura, *Interactive animation of virtual humans based on motion capture data*, Comput. Animat. Virtual Worlds **20** (2009), no. 5-6, 491–500, ISSN 1546-4261.

[MM05] Michael Meredith and Steve Maddock, *Adapting motion capture data using weighted real-time inverse kinematics*, Comput. Entertain. **3** (2005), no. 1, 5–5, ISSN 1544-3574.

[NNT08] Ryunosuke Nishimoto, Jun Namikawa, and Jun Tani, *Learning Multiple Goal-Directed Actions Through Self-Organization of a Dynamic Neural Network Model: A Humanoid Robot Experiment*, Adaptive Behavior **16** (2008), no. 2-3, 166–181, ISSN 1059-7123.

[RP06] Romesh Ranawana and Vasile Palade, *Multi-Classifier Systems: Review and a roadmap for developers*, Int. J. Hybrid Intell. Syst. **3** (2006), no. 1, 35–61, ISSN 1448-5869.

[WAH$^+$10] Kevin Wampler, Erik Andersen, Evan Herbst, Yongjoon Lee, and Zoran Popović, *Character animation in two-player adversarial games*, ACM Trans. Graph. **29** (2010), no. 3, ISSN 0730-0301, Article no. 26.

[WDA$^+$12] Zhikun Wang, Marc Deisenroth, Heni Ben Amor, David Vogt, Bernhard Schoelkopf, and Jan Peters, *Probabilistic Modeling of Human Movements for Intention Inference*, Proceedings of Robotics: Science and Systems (R:SS), 2012.

[YL10] Yuting Ye and C. Karen Liu, *Optimal feedback control for character animation using an abstract model*, ACM Trans. Graph. **29** (2010), no. 4, 74:1–74:9, ISSN 0730-0301.

[YT08] Yuichi Yamashita and Jun Tani, *Emergence of Functional Hierarchy in a Multiple Timescale Neural Network Model: A Humanoid Robot Experiment*, PLoS Comput Biol **4** (2008), no. 3, e1000220, ISSN 0730-0301.

[Zie96] Tom Ziemke, *Radar Image Segmentation using Recurrent Artificial Neural Networks*, Pattern Recognition Letters **17** (1996), no. 4, 319–334, ISSN 0167-8655.

Citation

David Vogt, Heni Ben Amor, Erik Berger, and Bernhard Jung, *Learning Two-Person Interaction Models for Responsive Synthetic Humanoids,* Journal of Virtual Reality and Broadcasting, 11(2014), no. 1, January 2014, urn:nbn:de: urn:nbn:de:0009-6-38565, ISSN 1860-2037.