A Data-Driven Method for Real-Time Character Animation in Human-Agent Interaction

David Vogt[1], Steve Grehl[1], Erik Berger[1],
Heni Ben Amor[2], and Bernhard Jung[1]

[1] Institut für Informatik,
Technische Universität Bergakademie Freiberg,
Bernhard-von-Cotta Str. 2, 09599 Freiberg, Germany
erik.berger@informatik.tu-freiberg.de
http://www.informatik.tu-freiberg.de

[2] Institute for Robotics and Intelligent Machines,
Georgia Institute of Technology,
801 Atlantic Drive, Atlanta, GA 30332-0280, USA
hba7@mail.gatech.edu
http://robotics.gatech.edu/

Abstract. We address the problem of creating believable animations for virtual humans that need to react to the body movements of a human interaction partner in real-time. Our data-driven approach uses prerecorded motion capture data of two interacting persons and performs motion adaptation during the live human-agent interaction. Extending the interaction mesh approach, our main contribution is a new scheme for efficient identification of motions in the prerecorded animation data that are similar to the live interaction. A global low-dimensional posture space serves to select the most similar interaction example, while local, more detail-rich posture spaces are used to identify poses closely matching the human motion. Using the interaction mesh of the selected motion example, an animation can then be synthesized that takes into account both spatial and temporal similarities between the prerecorded and live interactions.

Keywords: character animation, interaction mesh, virtual agent, interactive characters.

1 Introduction

Intelligent virtual agents have found widespread applications ranging from computer games [1,2], to educational software [3,4], or even shopping assistants [5]. Advances in sensing and graphics technology have significantly affected the development of avatars and their acceptance by users. An important example for this development is the introduction of affordable, low-cost motion tracking cameras. Instead of relying on artificial interfaces between humans and virtual agents, e.g. graphical user interfaces or joysticks, we can now directly analyze the user's body movement and thereby allow for a much more natural interaction.

T. Bickmore et al. (Eds.): IVA 2014, LNAI 8637, pp. 463–476, 2014.

Fig. 1. A virtual agent's animation is calculated based on live human motion data. For that we first analyze the user's current and previous postures to select the interaction he's currently in. After matching temporal as well as spatial aspects we further optimize the character's response with interaction meshes. In this way, a virtual agent can react to ongoing human motions for different interactions in real-time.

However, the ability to track human motion also introduces two key challenges to the design of intelligent avatars: (1) recorded motion needs to be classified and (2) adequate responses by the avatar need to be triggered. A prevalent approach to solving this task is to use machine learning algorithms in order to identify the semantics of a recorded movement and then trigger a recorded movement or behavior as a response [6,7]. Such classification-based approaches typically require the human movement to be seen entirely before a response can be triggered. In addition, they often do not generalize to different variations of the movement.

In this paper, we investigate a method for human-agent interaction which allows a virtual agent to react to ongoing human motions interactively and in real-time. Towards this end, we propose a data-driven approach to reactive motion generation based on motion capture data and multivariate time series analysis. First, human-human interactions are recorded in order to create a library of appropriate responses. Then, during human-agent interactions, the user's live motion is analyzed in segmented low-dimensional spaces. In doing so, we identify suitable responses by aligning the observed motion to the templates in our library. The resulting response is then optimized utilizing *Interaction Meshes* [8] to allow for fine grade adaptation to the observed human motion. We extend the Interaction Mesh approach by adding a context-aware decision layer that allows multiple two-person interactions to be triggered. Temporal contexts are embedded implicitly in the low-dimensional space which enable temporal drifts and varying motion speeds.

The remainder of the paper is organized as follows. In section 2, we review existing techniques and outline key advantages of our approach. In section 3, we introduce the concept of global and local posture spaces and describe the motion matching algorithm. Consequently, we present an algorithm for interactive generation of responses for virtual characters. In section 4, we present different experiments that were performed using an human-sized virtual character in an immersive CAVE environment.

2 Related Work

Enabling a virtual human to engage in interactions with users that involve complex motions has long been a goal of researchers. Towards that end different example-based approaches have been proposed. In [9] for example the authors propose a framework for online action recognition using histograms. In doing so, they map live motions to existing motion capture datasets with a dynamic timewarp implementation. The temporal context of postures is preserved by using dynamic programming.

Camporesi et al. describe a data-driven method for virtual agent animation based on demonstrated user motions [10]. For that several examples are recorded using a motion capture system with a reduced set of markers. During runtime character movements are computed by blending example motions to new situations. The underlying minimization problem is solved efficiently which allows for real-time motion generation.

Naour et al. [11] propose a method based on Laplacian mesh editing [12], which has proven to be well suited for animating close interactions. Here interaction meshes [13] are defined with respect to their temporal correlations in the original animation. Motion optimization is achieved by solving two minimization problems. The first penalizes deformations that results from Laplacian coordinate manipulation and the second preserves the length of motion segments over time.

The approaches presented so far allow interactive motion manipulation of a virtual character, however, live human motions are usually not incorporated and, thus human-agent interactions are not supported.

In contrast to that, Taubert et al. [14] presented an approach based on a hierarchical probabilistic model (GP-LVMs with a GPDM on top) to capture recorded motions for live human-agent interactions. During runtime different emotional styles of movements can then be synthesized. However, due to the computational expensive nature of the underlying probabilistic algorithm, model learning is a time consuming process.

Ho et al. [8] propose a method based on two-person motion capture data with a two stage process. First, the postures of the afterwards active interaction partner, i.e the human, and the virtual agent are organized in a kd-tree. This leads to a tree where each leaf stores pairs of poses that have been obtained in the initial recording. Then, for live human-agent interactions the tree is queried for postures that are similar to the current user pose. Here, Euclidean distances among 15-dimensional feature sets, in particular positions of both hands, feet and the pelvis are used as similarity measure. The most similar pose leads then, to the corresponding frame of the initial recording. After that, the characters pose is morphed to match spatial constraints by solving a space-time optimization problem with interaction meshes [13]. In doing so a virtual character can react to an ongoing human interaction in real-time.

However, in their approach Ho et al. synthesize a character's motion solely based on a single user posture and the last neighbor search result. This limits the temporal context of the response generation to two poses since long-range temporal dependencies are not considered. In order to distinguish between interactions that share similar segments, larger memories have to be taken into account to animate the character in a believable/contextual manner. E.g., towards the end of many interactions, one agent returns to the rest pose while the other agent is following-through an interaction-specific finishing motion. To that end, we propose the usage of an interaction memory as well as motion segments rather then single postures to allow wider temporal coherences.

Also, during runtime Ho et al. use 15 dimensions corresponding to 5 Cartesian positions, i.e. both hands, feet and the pelvis to match the current user posture in their kd-tree. However, for complex motions like dancing elbows, knees and the head are often a crucial part of the dance style. In our approach, we propose low-dimensional posture spaces [15] which capture intrinsic information that constitutes the essence of each interaction.

3 Methodology

The goal of this work is to provide a motion synthesis framework that allows virtual characters to engage in two-person interactions with a human counterpart. Our data-driven approach is based on the observation and generalization of a recorded library of human-human interactions. We first record two-person interactions with a motion capture system[1]. The recorded interactions encapsulate information on how the movements of each person affect behavioral responses from his partner. Leveraging this information, we propose an algorithm that automatically identifies the appropriate reaction to be performed by a virtual agent during human-agent interaction.

Our approach is based on the concept of low-dimensional posture spaces [15]. Low-dimensional posture spaces capture the intrinsic correlation among joint positions during motor skill generation. As a result, they can be used to generalize recorded movements to new, unobserved situations while at the same time reducing model complexity. In this paper we show, that movements during two-person interactions can also be captured by a low-dimensional posture space, e.g. the joint movement lies on a low-dimensional manifold.

The proposed method uses dimensionality reduction to extract important correlations during joint behavior. Figure 2 shows an overview of the approach.

[1] In our experiments we use a A.R.T DTrack 2 motion capture system.

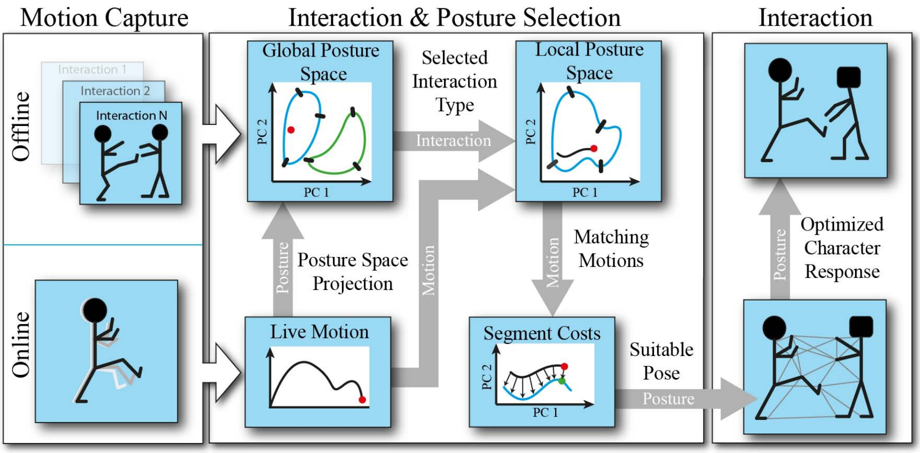

Fig. 2. Overview of the proposed method. First, we record two-person interactions with a motion capture system. Then the data is reduced in dimensionality leading to a global posture space as well as a set of local posture spaces specific to each interaction. To calculate a suitable character response during the online interaction, a reference frame of the original motion has to be identified by first using the global posture space to select an interaction and second by searching for similar motions in the corresponding local posture space. This leads to a pair of poses, i.e. postures of both interactants during the initial recording, which is then used to calculate an optimized posture for the virtual character.

The following description provides a brief overview of the steps involved in our method:

Posture Space Projection The user's live motion is captured and projected into the low-dimensional global posture space.

Selecting an Interaction Type Given the movements in the low-dimensional global interaction space, we identify the interaction type that fits the observation best.

Identifying Matching Motions User postures are projected into the local posture space of the active interaction type. Similarities between the projected trajectory and recorded movements from the library of training data are calculated. The segment with the highest similarity, i.e., a motion with similar postural changes is selected.

Extracting a Suitable Pose A cost matrix which captures distances to the best fitting motion segment is calculated to identify a point, i.e., a pair of poses of the initial recording, which satisfies postural as well as temporal requirements.

Optimizing a Character's Response Since the user's motion will vary as compared to the initial recording, we optimize selected character poses to the new situation by using interaction meshes [13].

Throughout the following sections we will focus on each step in detail. First, we will introduce global and local posture spaces.

3.1 Creating Posture Spaces

It is known, that motion capture data can be treated as a multivariate time series and is intrinsically based on low-dimensional manifolds [15]. Hence, one can employ dimensionality reduction techniques such as *Principal Component Analysis (PCA)* to strip off redundant information. When applied to motion capture data, each point in the resulting low-dimensional space corresponds to a posture and, hence, a trajectory to a motion when projected back to its original dimension.

In the recorded interactions, one interaction partner assumes an *active* role, while the other interactant has an *reactive* role. During the live human-agent interaction, the human is the active partner. We apply PCA to the motion capture data of the active interactant in the demonstrated interactions in several ways. First, we concatenate all motion capture recordings of the active interactant and project them into a single low-dimensional space, called *Global Posture Space* $\mathcal{I}$. Here all poses $x^{\mathcal{I}}_{i,n}$ are reduced to k dimensions. Second, for each demonstrated interaction i a l-dimensional *Local Posture Space* $\mathcal{P}_i$ is calculated to capture small details in the user's motion. To select a suitable amount of principal components k and l respectively we only select Eigenvectors that encode an entropy above 1 percent [15]. Hence, each posture space has a different dimensionality ensuring that enough dimensions are used to preserve small details of motions while at the same time reducing computational costs.

However, if we would consider the motion as a whole, alternations in the relationship of latent variables would not be taken into account [16]. Therefore, we split each interaction type in j segments, using Hotelling's T^2 statistics, which tries to capture changes of the underlying correlation structure. It does so, by minimizing the variance over a segment $\mathcal{S}_{i,j}$ and, consequently, concatenating temporal consecutive postures to motion segments in low-dimensional space. The cost function can be formalized as follows.

$$\mathcal{S}_{i,j} := \{x_{i,n=a}, x_{i,n=a+1}, \ldots, x_{i,n=b}\} \tag{1a}$$

$$cost_{T^2}(\mathcal{S}_{i,j}) = \frac{1}{b-a+1}\sum_{j=a_i}^{b_i} T^2_{i,j} \qquad T^2_{i,j} = x^T_{i,n} x_{i,n} \tag{1b}$$

The segmented global posture space is now used to select an interaction template in a ongoing human-agent interaction.

3.2 Selecting an Interaction Template

To classify an ongoing human agent interaction the user's posture $y^{\mathcal{H}}$ is projected into the global posture space $\mathcal{I}$ leading to a new point $y^{\mathcal{I}}$. The Euclidean distances d_i between this point and the closest segment centroid of each interaction i are calculated and added to a so called *interaction memory* D. This is done for the current time step t and the previous q time steps.

$$D = (d_1 \; d_2 \ldots d_i \ldots d_m) \; , \; d_i = \left(d_i^t \; d_i^{t-1} \; \ldots \; d_i^{t-q}\right)^T \tag{2a}$$

$$d_i = min(\|y^{\mathcal{I}} - centroid(\mathcal{S}_{i,j}^{\mathcal{I}})\|) \; , i \in \mathbb{N} : [1, m] \tag{2b}$$

Every column i captures the distances to the closest centroid of interaction i. Each row stores the distances of a single time step for all interactions. The interaction with the smallest mean value over all q rows identifies an interaction template.

3.3 Identifying Matching Motions

For believable human agent interactions a character's response highly depends on temporal features of the interaction. In a dance motion for example, it is crucial to be in strict time whereas for a high five movement the interactants' hands have to meet at the right time. In order to allow a virtual character to engage in such interactions past user poses have to be utilized. For that, a set of previous poses is projected into the local posture space of the active interaction, leading to a new motion trajectory $y_o^{\mathcal{P}_i}$.

Then, a similarity value s_{PCA} is calculated for each neighboring posture space segment. By only including the neighborhood, we restrict possible character responses to motions that contain postures similar to the ones obtained in the initial recording. The comparative measurement is defined as the sum of angles between each pair of sub principal components which can be formalized as [17]:

$$s_{PCA} = \frac{1}{l} \sum_{k=1}^{l} \sum_{c=1}^{l} cos^2 \Theta_{kc} = \frac{1}{l} trace(U_{k,l}^T U_{c,l} U_{c,l}^T U_{k,l}) \tag{3}$$

Here, l is the number of principal components for local comparison. Θ_{kc} denotes the angle between k^{th} principal component (PC) of a segment and the c^{th} sub PC of the user motion. In order to calculate the similarity value efficiently, the equation is reformulated in matrix form where $U_{k,l}$ is the subspace defined by the eigenvectors of the covariance matrix for segment k with dimension l.

In essence, the algorithm assigns high similarity values for segments with PC axes pointing in the same direction. In other words, segments with similar postural changes over time are assigned large similarity values. The segment j that fits the last user motion best is now used for further optimization.

3.4 Extracting a Suitable Pose

Since segment lengths vary for different motion parts of an interaction the following accumulated cost matrix is employed to identify a temporal matching pose of

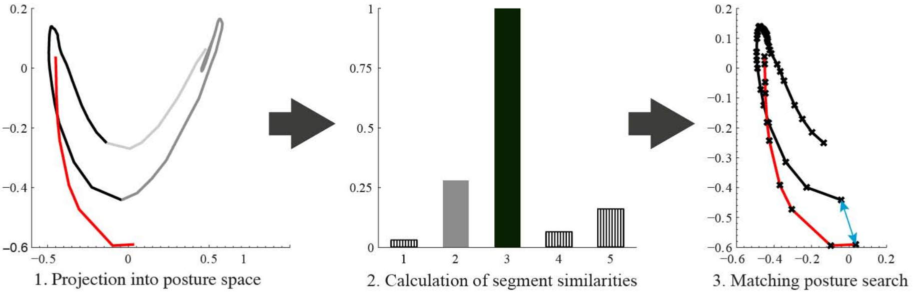

Fig. 3. The figure shows all required steps to identify a frame of the initial motion recording for an ongoing human-agent interaction. First the live user movement is projected into low-dimensional space leading to a new trajectory. After that, most similar segments are labeled utilizing Hotelling's T^2 statistics. Finally, the most similar posture is extracted by evaluating a cost matrix.

the original recording [18]. In doing so, we evaluate distances in low-dimensional space of live user poses $y_o^{\mathcal{P}_i}$ and template postures $x_{i,n}^{\mathcal{P}_i}$.

$$S_{n,o} := \begin{cases} \sum_{n=a}^{b} \|y_1^{\mathcal{P}_i} - x_{i,n}^{\mathcal{P}_i}\| & \forall n \in \mathbb{N} :]a,b],\ o = 1 \\ \|y_o^{\mathcal{P}_i} - x_{i,a}^{\mathcal{P}_i}\| & \forall o \in \mathbb{N} :]1,O],\ n = a \\ min\{S_{n-1,o-1}, S_{n,o-1}, S_{n-1,o}\} & \\ \quad + \|y_o^{\mathcal{P}_i} - x_{i,n}^{\mathcal{P}_i}\| & \forall n \in \mathbb{N} :]a,b],\ o \in \mathbb{N} :]1,O] \end{cases} \tag{4}$$

To summarize, figure 3.3 shows required steps to select a frame of the initial recording that fits the motion context of the selected interaction template while at the same time satisfying postural similarities. The resulting pose is now subject to spatial optimization using interaction meshes.

3.5 Optimizing a Characters Response

In the previous step a frame of the initial recording has been identified that represents the most similar user posture. However, the user's movement will differ from the original motion in form and size and, thus, further optimization is required. To retain the characteristics of the prerecorded interaction, we optimize the selected posture by using interaction meshes [13]. In doing so, we minimize the Laplacian deformation energy [12] of a newly created mesh with regard to the one created during the initial recording. Here, the Laplacian deformation energy is defined as follows:

$$E_L(V) := \sum_j \frac{1}{2} \|L(x_j) - L(y_j)\|^2 \tag{5}$$

L is the operator to compute the Laplacian coordinates from given Cartesian coordinates $V = (p_1, \ldots, p_{2m})^T$. x_j are vertex locations (motion capture markers) from the prerecorded motion capture data in the selected frame, whereas y_j are coordinates of markers from the live human-agent interaction. Laplacian coordinates of a vertex are obtained as follows:

$$L(p_j) = p_j - \sum_{l \in N_j} w_l^j p_l \tag{6}$$

N_j is the one-ring neighborhood of vertex p_j.
Since we want to react to an ongoing user motion, additional positional constraints have to be defined on the interaction mesh.

We treat the current user posture as a hard constraint in our optimization problem. Additionally, we define soft constraints on the character vertices to retain further desired aspects of the interaction, e.g. supporting foot contact and body position. The resulting optimization problem subject to the soft and hard constraints can be reformulated as system of linear equations (cf. [13]):

$$\begin{bmatrix} \mathcal{M}^T\mathcal{M} + F^T\mathcal{W}F & \mathcal{C}^T \\ \mathcal{C} & 0 \end{bmatrix} \begin{bmatrix} \mathcal{V} \\ \lambda \end{bmatrix} = \begin{bmatrix} \mathcal{M}^T\mathcal{B} + \mathcal{F}^T\mathcal{W}f \\ h \end{bmatrix} \tag{7}$$

where $\mathcal{V}$ and λ denote the vertices of the deformed interaction mesh and the Lagrange modifiers respectively. $\mathcal{M}$ is the Laplacian matrix of the original motion. $\mathcal{C}$ is the matrix of all constraints which can be separated into the matrix $\mathcal{F}$ of soft constraints, e.g. the virtual agent's position constraints, and the vector h of hard constraints, e.g. the user's current posture. Each soft constraint f is weighted by the weight matrix $\mathcal{W}$. $\mathcal{M}^T\mathcal{B}$ denotes the transformation of the original vertex positions $\mathcal{B}$ in Cartesian coordinates into Laplacian coordinates.

A solution of the system of linear equations is an interaction mesh $\mathcal{V}$ that minimizes the Laplacian deformation energy while satisfying the different constraints. However, vertex locations cannot be transferred to a virtual character without further post-processing, since not all joints correspond to a vertex. In order to calculate rotations for each bone we utilize an inverse kinematics solver.

4 Evaluation

To evaluate our method we recorded two-person interactions, namely high five, a hand clapping game, waving at each other and a jive dance. The corresponding 15 dimensional global posture space is illustrated for the first 3 principal components in figure 4. In a live human-agent interaction a user was tasked to high five the virtual agent. As expected its motion varied from the initial recording, however, its trajectory in low-dimensional space stills followed the same direction. This is due to the fact that similar postures were adopted which in turn lead to neighboring low-dimensional points.

On the right hand side of figure 4 the local posture space of the selected interaction is visualized. Here the closest matching motion segment of the initial

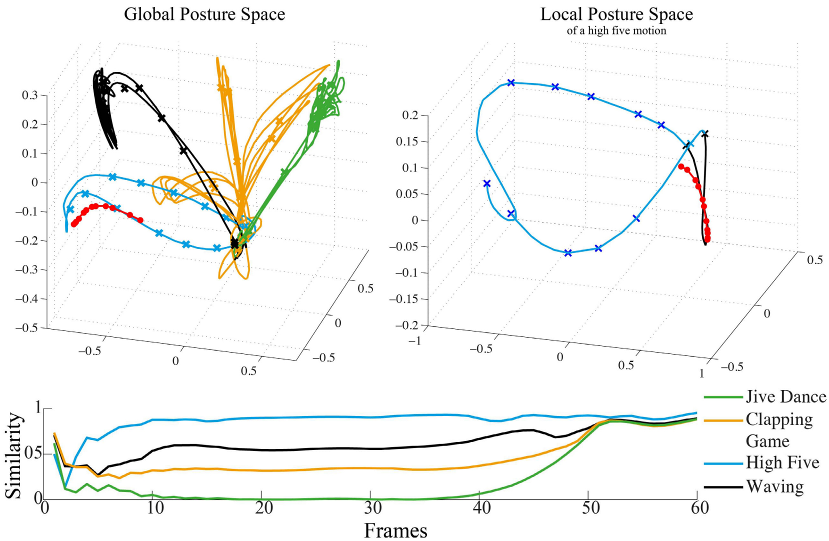

Fig. 4. In the figure on top the global as well as the selected local posture space is shown. A user was tasked to high five the virtual agent. Below the mean activation in our interaction memory is visualized. As can be seen, a recorded high five motion is most similar to the executed user motion. However, one can also conclude that other motions also exhibit similar poses especially around frame 50 to 60.

Fig. 5. The virtual agent's postures are optimized for live human agent interactions. In this example a user high fives a virtual character successfully. The agent adopts its motion to meet the users hand at the right time and position.

recording is marked (black trajectory). As can be seen, the user motion (indicated by the red trajectory) also follows the path of the closest segment. After calculating the cost matrix, a matching point is selected and its associated interaction mesh is optimized. The resulting character responses can be seen in figure 5 for 3 frames.

In a second example we utilized the same global posture space to detect a ongoing jive dance motion. The projection of current and recent user postures into the global local low-dimensional spaces are shown in figure 6 top left. As can be seen, its motion matches the shape of the jive template which has been generated from the initial recording. Additionally, the local posture space corresponding to

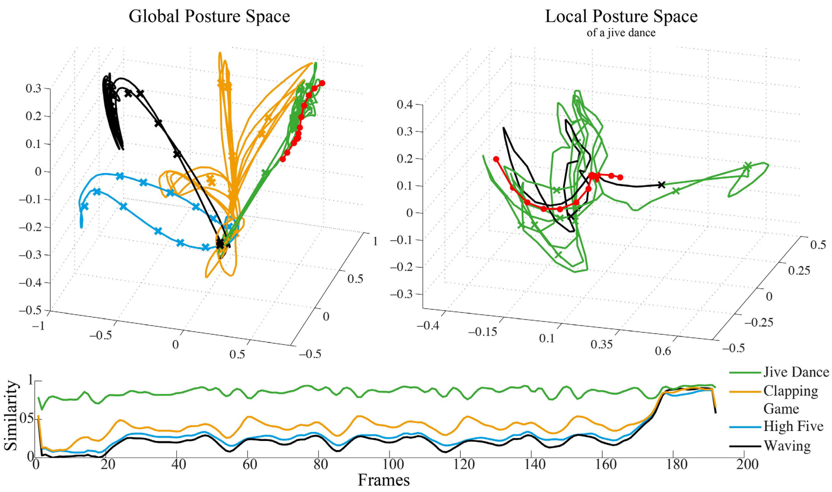

Fig. 6. The global as well as the selected local posture space of a jive dance motion are shown on top. The live user postures are highlighted red. The mean activation for each interaction is outlined below.

Fig. 7. The motion of a virtual character is optimized in real-time using our interaction learning method. As can be seen the agent successfully imitates the behavior shown in the initial recording.

the selected interaction is shown. The most similar segment is highlighted. The final character response can be seen in figure 7.

The similarities of the live user motion to recorded interaction examples are in figure 6 bottom. The reason for the large similarities towards the end of the interactions is that in all our recordings, the active participants returned to a pose with both arms resting aside.

In a third example a hand clapping game is performed with a virtual character. Here the same global posture space that has been created from the initial recordings is used. As shown in figure 8 the projected user postures (highlighted red) match the template created from a clapping game motion. However, as illustrated below the selected interaction type has been a high five at first (see frame 1 to 20) but changed later to the correct interaction. The reason for that

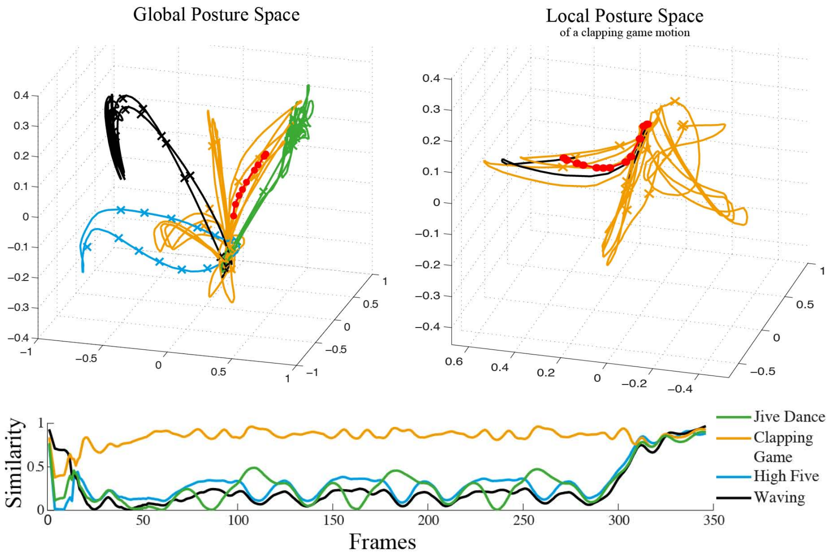

Fig. 8. The figure shows the global posture space and projected live user postures (highlighted red). On the right hand side the selected local posture space of a clapping game motion is visualized with 10 previous user poses for motion matching. Additionally, the similarities for each interaction type are illustrated below. As can be seen a high five motion is selected at first but changed later to the correct clapping game.

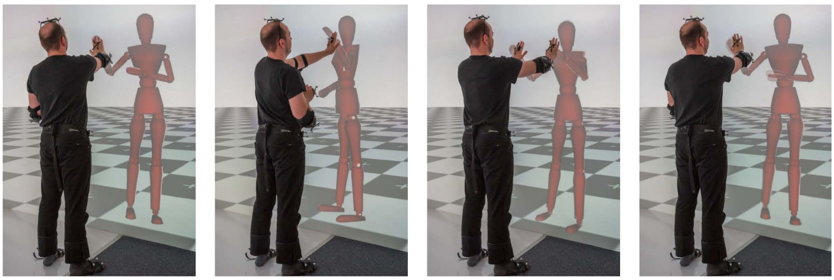

Fig. 9. With our method a virtual character can respond to complex interactions like a clapping game as shown in the figure for 4 key postures. Here the agent's hand has to meet the users palm at the right time and at the right position.

are similar postures that have been obtained in both motion capture recordings. The final character responses are illustrated for 4 key postures in figure 9.

5 Discussion

The presented approach utilizes a so called interaction memory for hysteresis effects and to allow a virtual agent to remain in an interaction. As a result it can potentially lock the agent in one interaction. The sliding window size of the interaction memory obviously affects the overall latency of the system. In our experiments a memory size q of 10 has been proven to be well suited. This leads to memory length of approximately 0.5 seconds at 20 frames per second and to a lag of 300ms on average on a modern Macbook Pro. Selecting a posture in global posture space as well as motion matching in local posture spaces takes on average 0.008s whereas optimizing the interaction meshes itself takes 0.01 seconds. In our current implementation, transforming the resulting vertex coordinates to joint angles utilizing the inverse kinematics solver takes twice as long (0.029 seconds).

Currently, the proposed method does not allow for additional objects to be included in two-person interactions as global as well as local posture spaces are not sensitive to object ownerships. Furthermore, we currently do not track, and thus cannot recreate the interactants' hand shapes during interactions.

6 Conclusion

In this paper, we presented a new, data-driven method for generating real-time responses of an interactive virtual human. Using training data acquired from human-human interactions, we generate low-dimensional representations that allow for the generalization of the observed behavior to different variations thereof. In doing so, crucial characteristics of an interaction as well as small details of motions are preserved and used to animate a virtual agent. We extended the approach presented by Ho et al. [8] to situations where the temporal context of interactions plays an important role.

Experiments performed in an immersive virtual environment show that the approach can be used for synthesizing context-aware responses in real-time. As a result, a more natural interaction between a virtual agent and a human user can be established.

As a possible extension of the approach, we are currently considering the implementation of time-varying interaction meshes as well as a probability based segmentation to allow for overlapping segments. In addition we are also investigating the use of the proposed approach in the generation of robot responses during human-robot interaction.

References

1. Deng, L., Leung, H., Gu, N., Yang, Y.: Real-time mocap dance recognition for an interactive dancing game. Computer Animation and Virtual Worlds 22(2-3), 229–237 (2011)
2. Kim, J., Seol, Y., Lee, J.: Human motion reconstruction from sparse 3d motion sensors using kernel cca-based regression. Computer Animation and Virtual Worlds 24(6), 565–576 (2013)
3. Weimin, X., Wenhong, X.: E-learning assistant system based on virtual human interaction technology. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) ICCS 2007, Part III. LNCS, vol. 4489, pp. 551–554. Springer, Heidelberg (2007)

4. Osterlund, J., Lawrence, B.: Virtual reality: Avatars in human spaceflight training. Acta Astronautica 71(0), 139–150 (2012)
5. Chattaraman, V., Kwon, W.-S., Gilbert, J.E.: Virtual agents in retail web sites: Benefits of simulated social interaction for older users. Computers in Human Behavior 28(6), 2055–2066 (2012)
6. Jung, B., Amor, H.B., Heumer, G., Vitzthum, A.: Action capture: A vr-based method for character animation. In: Coquillart, S., Brunnett, G., Welch, G. (eds.) Virtual Realities, pp. 97–122. Springer, Heidelberg (2008)
7. Bouchard, D., Badler, N.I.: Semantic segmentation of motion capture using laban movement analysis. In: Pelachaud, C., Martin, J.-C., André, E., Chollet, G., Karpouzis, K., Pelé, D. (eds.) IVA 2007. LNCS (LNAI), vol. 4722, pp. 37–44. Springer, Heidelberg (2007)
8. Ho, E.S.L., Chan, J.C.P., Komura, T., Leung, H.: Interactive partner control in close interactions for real-time applications. ACM Trans. Multimedia Comput. Commun. Appl. 9, 21:1–21:19 (2013)
9. Barnachon, M., Bouakaz, S., Boufama, B., Guillou, E.: Ongoing human action recognition with motion capture. Pattern Recognition 47(1), 238–247 (2014)
10. Camporesi, C., Huang, Y., Kallmann, M.: Interactive motion modeling and parameterization by direct demonstration. In: Allbeck, J., Badler, N., Bickmore, T., Pelachaud, C., Safonova, A. (eds.) IVA 2010. LNCS, vol. 6356, pp. 77–90. Springer, Heidelberg (2010)
11. Le Naour, T., Courty, N., Gibet, S.: Spatiotemporal coupling with the 3d+t motion laplacian. Computer Animation and Virtual Worlds 24(3-4), 419–428 (2013)
12. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.-P.: Laplacian surface editing. In: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP 2004, pp. 175–184. ACM, New York (2004)
13. Ho, E.S.L., Komura, T., Tai, C.-L.: Spatial relationship preserving character motion adaptation. ACM Trans. Graph. 29 (July 2010)
14. Taubert, N., Löffler, M., Ludolph, N., Christensen, A., Endres, D., Giese, M.A.: A virtual reality setup for controllable, stylized real-time interactions between humans and avatars with sparse gaussian process dynamical models. In: Proceedings of the ACM Symposium on Applied Perception, SAP 2013, pp. 41–44. ACM, New York (2013)
15. Ben Amor, H.: Imitation Learning of Motor Skills for Synthetic Humanoids. PhD thesis, Technische Universität Bergakademie Freiberg (2010)
16. Bankó, Z., Abonyi, J.: Correlation based dynamic time warping of multivariate time series. Expert Systems with Applications 39(17), 12814–12823 (2012)
17. Krzanowski, W.J.: Between-groups comparison of principal components 74, 703–707 (September 1979)
18. Vlachos, M., Hadjieleftheriou, M., Keogh, E.J., Gunopulos, D.: Indexing multidimensional trajectories for similarity queries. In: Spatial Databases, pp. 107–128 (2005)