In Proceedings of Intelligent Virtual Agents (IVA), Philadelphia, Pennsylvania, 2010

Interactive Motion Modeling and Parameterization by Direct Demonstration

Carlo Camporesi, Yazhou Huang, and Marcelo Kallmann

University of California, Merced

Abstract. While interactive virtual humans are becoming widely used in education, training and therapeutic applications, building animations which are both realistic and parameterized in respect to a given scenario remains a complex and time–consuming task. In order to improve this situation, we propose a framework based on the direct demonstration and parameterization of motions. The presented approach addresses three important aspects of the problem in an integrated fashion: (1) our framework relies on an interactive real-time motion capture interface that empowers non–skilled animators with the ability to model realistic upper-body actions and gestures by direct demonstration; (2) our interface also accounts for the interactive definition of clustered example motions, in order to well represent the variations of interest for a given motion being modeled; and (3) we also present an *inverse blending* optimization technique which solves the problem of precisely parameterizing a cluster of example motions in respect to arbitrary spatial constraints. The optimization is efficiently solved on-line, allowing autonomous virtual humans to precisely perform learned actions and gestures in respect to arbitrarily given targets. Our proposed framework has been implemented in an immersive multi-tile stereo visualization system, achieving a powerful and intuitive interface for programming generic parameterized motions by demonstration.

Keywords: Learning by demonstration, motion blending, virtual humans, virtual reality.

1 Introduction

A central goal in the area of autonomous virtual humans is to achieve virtual assistants that can effectively interact, learn, train, and assist people in a variety of tasks. We focus on the particular problem of modeling motions for interactive training applications requiring complex gestures and actions to be reproduced realistically and with precise parameterizations in respect to spatial constraints in the environment.

Modeling and parameterization of realistic motions is clearly an important problem in a wide range of applications involving virtual humans. One approach for achieving precise parameterizations is to rely on algorithmically synthesized actions and gestures [13, 19], however it remains difficult to achieve realistic full-body results and a specific computational model is needed for every action to be simulated.

Another important limitation of algorithmic approaches in many training applications is that the motions to be reproduced may only be known by experts in the subject

area of the training. Such cases are clearly better handled by motion modeling solutions based on motion capture.

Several systems based on motion captured (or hand-crafted) animations have been developed and are able to achieve highly realistic results [8, 28]. However the process of building the set of needed motions for each given scenario is often time–consuming, and it remains difficult to precisely parameterize pre-defined animations in respect to spatial constraints.

We propose in this paper an interactive motion modeling framework for addressing these many difficulties in an integrated fashion. Our framework is designed to be used in two distinct phases: in the **modeling phase** the user demonstrates to the virtual human how to perform parameterized motions, such that in the **training phase** the virtual human is then able to reproduce the motions in interactive training sessions with apprentice users learning the training subject.

Our system targets the situation where, in the modeling phase, experts in the training subject are able to model the needed actions and gestures by direct demonstration, without the need of having previous experience with the system. In the training phase, the stored example motions are then re-used by the virtual human to train apprentice users. Our framework in particular enables the virtual human to reproduce motions in respect to arbitrary target locations in the environment. Figure 1 presents one typical scenario modeled by our system.

(a) (b)

Fig. 1. (a) In this scenario, during the modeling phase, the user demonstrates several examples of pointing motions in order to demonstrate operations with a stereo system and a telephone. (b) During the training phase, the user requests the virtual human to precisely perform the same pointing motions for arbitrary targets, here specified by the apex of the yellow cone which is controlled via a WiiMote controller. The training phase is used here to test if the example motions are sufficiently covering the volume of interest in the scenario. The user can interactively switch between the two phases until all required motions are correctly defined. Note that the simulated images in this figure appear fuzzy since they are being projected for stereo visualization.

Our motion-based interactive framework allows the design of new types of interaction techniques, which can be developed according to the training scenario at hand. For example, the apprentice may request the virtual human to perform actions at different locations and under different conditions, feedback can be provided based on on-line

comparisons between the motions from the expert and the apprentice, etc. Such scenarios are clearly applicable to a variety of training applications, for example, sports training, rehabilitation of motor-impaired patients, training of medical procedures, demonstration of generic procedures, delivery of instructions, etc.

The work presented in this paper addresses the main computational challenges involved in building such interactive systems. Our proposed framework is based on three main computational modules:

- First, a real-time motion capture interface is developed for allowing users to interactively model motions by direct demonstration. In order to be effective for a variety of situations, our solution includes calibration and mapping from a reduced set of tracking sensors.
- During the motion modeling phase, motions can be recorded and re-played on-line, allowing users to effectively model generic actions and gestures in an intuitive way. The modeling interface also allows the definition of clusters of example motions, in order to represent spatial variations of a same motion. These variations are used during the training phase in order to precisely parameterize the motions in respect to arbitrarily given spatial targets.
- Finally, given a cluster of example motions built by the user during the modeling phase, we present an optimization technique for computing motions on-line precisely respecting arbitrarily given spatial constraints. Our technique is called *inverse blending* and the solution motions are obtained by blending operations with the example motions in a given cluster, therefore the solutions remain humanlike and similar to the example motions. This technique is critical for achieving precise parameterizations of realistic actions and gestures, without the need of any pre–computation. Examples are presented with the modeling of pointing and pouring motions, which are precisely parameterized in respect to arbitrarily given target locations.

Our interactive motion modeling framework has been implemented in an immersive multi-tile *power wall* stereo visualization system (shown in Figure 1). The ability to perform simulations in a large visualization system is important for achieving immersive full-scale interactions, in analogy to the way humans naturally interact with each other. As a result our obtained system represents a powerful and intuitive approach for programming generic parameterized motions by demonstration.

The reminder of this paper is organized as follows: after discussing related work in the next section, we present our motion capture interface in Section 3 and our modeling interface in Section 4. We then present our inverse blending optimization technique in Section 5. Finally, Section 6 discusses our results and Section 7 concludes this paper.

2 Related Work

Our approach of direct demonstration of motions is strongly related to several imitation-based learning methods previously proposed for different applications in robotics [2,26] and computer graphics [5,6]. The work of Cooper et al. [5] in particular also employs a full-body motion capture interface for building a database of motions, however with the

focus on building motion databases with good coverage for motion controllers, and not on achieving an immersive and interactive system to teach motions to virtual humans.

Although several existing systems address different aspects related to our work, we are actually not aware of other systems with all the same characteristics as ours. We therefore proceed with our related work analysis in respect to the different computational solutions employed in our overall system.

Several methods have been proposed in the literature for addressing the problem of motion reconstruction from a reduced marker set. A popular approach is to employ statistical models [31] and machine learning [4] for extracting from a motion database the motion closest to the input signal. The performance of these methods however greatly depends on the used databases and they are usually not suitable for real-time applications. Algorithmic approaches based on simulation or optimization have also been proposed [11, 24] but are computationally expensive and are not suitable for achieving humanlike motions.

Algorithmic approaches based on Inverse Kinematics (IK) can run in real-time and may be suitable for motion reconstruction if enough markers are provided to well limit the overall posture space of possible solutions. Inverse Kinematics has also been employed to optimize full–body postures for tracking the input stream from a reduced marker set [7, 20]. However the convergence time for iterative Jacobian-based solvers over the full–body of a character may require several iterations and can introduce lag in a real-time interface.

Our interactive interface focuses on achieving a fast solution for reconstructing humanlike motions by employing an analytical IK solver [12] applied only to the arms of the character. We then rely on simple mappings from additional markers in order to fully reconstruct upper-body motions very efficiently in real-time. As our present work focuses on modeling upper-body actions, we do not address in this work tracking of legs or reconstruction of locomotion.

One of the main purposes of our system is to model actions and gestures to be used in training applications. Previous work on gesture synthesis has mainly focused on sequencing pre-defined animations [8, 27, 28], or by algorithmic synthesis, such as by employing IK solvers towards specified trajectories [13, 19]. By modeling gestures with our motion blending techniques we are able to achieve the benefits of both approaches, i. e., realistic animations which can be also parameterized with spatial targets.

The topic of character animation based on motion capture has been extensively studied in the literature for several applications [1, 14–16, 25]. Although the majority of works focus on the locomotion problem, motion blending (or motion interpolation) has also been well addressed in previous works for modeling gestures and actions.

Different methods have been proposed related to motion blending, for example, hierarchical filtering [3], parameterization using Fourier coefficients [29], stochastic sampling [30], and interpolation based on radial basis functions (RBFs) [22]. The problem of end-effector control with motion blending has also been addressed before [23, 32], and more generically, spatial properties are also addressed by the geostatistical interpolation scheme [18].

Another approach used for addressing spatial constraints is to generate and add pseudo motion examples [14, 23], which however increases the needed computation and

storage. The scaled Gaussian process latent variable model [9] optimizes interpolation kernels specifically for maintaining constraints described by latent spaces.

The main limitation of these methods is that alone they are not able to precisely meet given spatial constraints. For instance, the active learning methodology [5] relies on Inverse Kinematics solvers in addition to blending, however risking to penalize the obtained realism.

Our proposed method for motion parameterization is based on the optimization of blending weights until best meeting generic spatial constraints defining the target parameterization. Our approach is simple and intuitive, and yet has not been addressed in previous work. Our method can be seen as a post-processing step for optimizing a given set of blending weights, which can be initially computed by any motion blending technique. Only error metrics for the spatial constraints to enforce are necessary in order to optimize the blending weights using a given motion interpolation scheme. We show in this paper that our optimization framework is able to well parameterize pointing and pouring actions on-line and without the need of any pre–computation.

3 Motion Capture Interface

A variety of commercially available solutions are able to map full–body motions to a virtual character in real-time, however the available options are usually expensive and often require the user to wear cumbersome tracking devices.

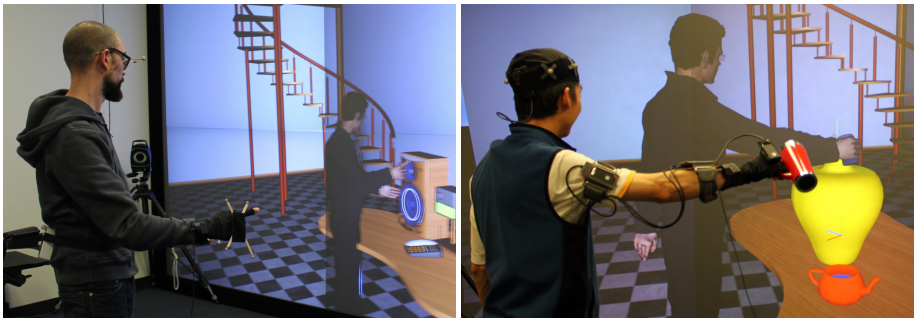
For instance, retro-reflective motion capture systems require the user to wear a full–body suit with a number of markers carefully placed; systems based on magnetic sensors rely on 15 to 20 magnetic sensors connected with cables; and exo-skeleton systems are heavy and often restrictive.

As shown in Figure 2 (a), our real-time upper-body motion capture solution is based on tracking four key limbs of the user: the two hands, the head, and the lower or mid joint of the spine. We track both the position and orientation of each of these parts in global coordinates. The user wears simple straps with markers on each of the considered body parts and we rely on a 10-camera Vicon system for performing the real-time tracking. Although we rely on an instrumented room with cameras, our solution can be ported to any other system able to track these four parts. We also rely on a data glove for capturing finger motions in real-time.

Before starting an interactive session, a calibration process is necessary in order to map the user's body to the skeleton of the virtual human in the scene. We choose not to adapt the dimensions of the virtual human in order to maintain a consistent database of motions which can be shared by different users.

The calibration consists of measuring scaling factors, and requires the user and the virtual agent to stand in a T–pose posture. Let e_i denote the positions of the hands and the head of the user in global coordinates, $i = \{1, 2, 3\}$. Let p be the global position of a point on the user spine at the same height as the shoulder. This point is computed from the spine and hand markers at T–pose. Similarly, let e_i^v and p^v be the same points but computed in respect to the virtual human skeleton. Scaling factors s_i are then computed with:

$$s_i = \frac{\|e_i^v - p^v\|}{\|e_i - p\|}. \tag{1}$$



(a) (b)

Fig. 2. The figure shows (a) the camera-based motion interface and (b) our *gesture vest* interface based on inertial sensors [10].

The scaling factors are obtained during the T-pose calibration and then applied during the modeling phase of the system in order to have the end-effector positions of the user and the virtual human matching. Each time new readings are processed, each scaling factor s_i multiplies the translation component of its corresponding body part being tracked, after transforming it to local coordinates in respect to the root joint.

The following additional operations are then performed on-line in order to complete the motion mapping:

- First, the global position of the virtual human is updated according to the tracked spine location of the user. In our current version, we focus only on capturing upper–body actions and gestures without any locomotion, and so no attention is given for tracking or solving leg motions.
- The spine of the virtual human is bent considering the amount of rotation between the tracked spine orientation and the head orientation. This rotation is subdivided by the number of joints being used to represent the spine of the virtual human, and distributed among these joints, similarly to the approach described by Monheit et al. [17]. The rotation of the head joint is directly mapped from the head markers, which are actually placed attached to the polarized glasses used for stereo visualization (see Figure 2 (a)).
- The arm posture reconstruction is performed using a fast analytical IK solver considering arm-body collision avoidance by automatic variation of the arm swivel angle [12]. The swivel angle is set to start at a low default value such that the elbows remain low, as is the case in usual arm motions. In some cases when the arm is extended, due tracking and calibration imprecisions, the IK may report that an end-effector cannot reach its target position, in which case we take the closest possible solution. The motion of the fingers is directly mapped from a data-glove. As shown in Figure 2 (a), four markers attached at the extremities of two crossed sticks (fixed on the data-glove) are used for tracking the 6 degrees of freedom hand targets.

Although a precise reconstruction cannot be guaranteed, the mapping is extremely fast and results in very fluid interactions always running well above 30 frames per sec-

ond. Achieving a fluid and lag-free interface has showed to be extremely important for the effective use of the system. The proposed use of a reduced marker set allows the accommodation of systems with fewer (or lower–cost) cameras and also allows the system to be ported to other tracking solutions.

We have in particular also experimented with an alternate motion capture solution based on our portable and easy–to–wear *gesture vest* prototype system [10]. This device is composed of 5 inertial sensors placed on the arms, head and spine of the user. The device produces very accurate motions, is portable and wireless, and is in particular well suited for capturing one-arm gestures. Figure 2 (b) illustrates one interactive modeling section using this equipment. This system is also integrated with a data glove in order to capture hand shapes.

The main drawback of this solution is that alone it cannot track information in global coordinates in respect to our visualization system, making it difficult to be used in our applications related to specification of spatial constraints. When the system is integrated with a global tracking device then it becomes perfectly suitable for our applications. Note that in Figure 2 (b) the user is wearing a hat being tracked by the camera system in order to provide global positioning information.

Depending on the application, and on the availability of additional trackers, our gesture vest solution represents a suitable alternative for achieving a small, portable and low–cost solution. The system can be in particular effective for large-scale training scenarios with many users, since the sensors scale well and do not suffer from occlusion limitations.

4 Interactive Motion Modeling Interface

The ability to run our system in integration with a large immersive display is of main importance in our interactive interface. It allows the user to interact with full-scale virtual environments with immersive stereo vision perception, achieving realistic and accurate reproduction of conditions during both the modeling and training phases.

In order to allow full operation of the system by a single user, our current solution for the motion modeling phase focuses on tracking single-arm actions and gestures performed by the right arm. In this way the user only wears a data glove on the right hand, and the left hand holds a WiiMote controller which provides control over all the system functionality, achieving a simple and effective interactive user interface.

By clicking buttons on the WiiMote controller, the user can change the camera view between several modes, can control the recording and replay of motions, initiate the definition of clustered example motions, add or delete motions from each cluster, etc.

The definition of clusters of example motions is an important concept of our system. The definition of a cluster is necessary for specifying each parameterized action or gesture. When the user selects to start a new cluster, every recorded motion becomes associated with the cluster. Motions in a cluster will be blended during the training phase and therefore they have to consistently represent variations of a same type of motion. For instance, a pointing cluster will contain several pointings of the same type but each pointing to a different location in the environment.

One important information to be associated to each motion in a cluster is its parameterization frame. For example, this frame will be the stroke frame of a gesture or the final frame of a one-way pointing motion. The parameterization frame identifies which frame of the motion is to be parameterized in respect to new target locations during the training phase. We currently let the user specify this frame by pressing a button of the WiiMote controller at the right moment with the left hand, while the motion is being demonstrated with the right arm. The frame location can be then adjusted forward and backwards interactively if needed. This solution is acceptable in several cases but we recognize it may divert the attention of the user from well performing the motion being demonstrated. We therefore also let the user to interactively select this frame after the motion is performed. We also allow the user to trim the initial and final frames of each recorded example motion.

Clusters of motions can be edited, stored and reloaded as needed. Whenever the user wishes to test a modeled cluster, the system can be switched to training phase and the WiiMote controller is then used to specify targets to be solved by inverse blending. In this way the virtual human is able to perform a new motion precisely reaching given targets and well preserving the quality of the original demonstrated motions. The WiiMote controller also has a haptic feedback which is used during the training phase to tell the user when asked targets are colliding with the environment.

The interconnection of the several modules of our system is further illustrated in Figure 3. In the next section we describe our inverse blending optimization technique used during the training phase.

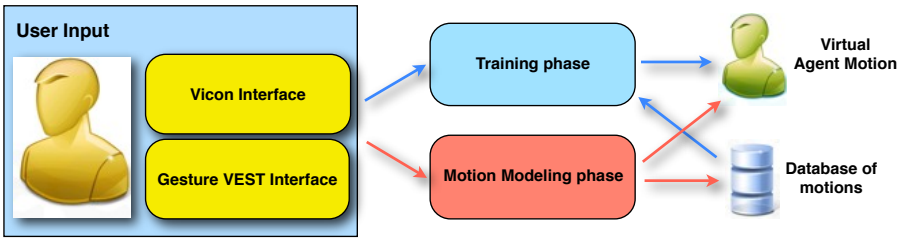


Fig. 3. Overview of the main parts of the system. The arrows illustrate the data flow during the modeling and training phases.

5 Inverse Blending

The first step for applying the inverse blending optimization is to model each spatial constraint of interest with an error metric function f, which measures how well each constraint is being satisfied at the given parameterization frame. Although the examples presented in this work only use positional constraints for end-effectors, generic types of spatial constraints C can also be taken into account.

Constraints can also have an arbitrary number of degrees-of-freedom (DOF), for example, pointing to a distant location imposes a 2–DOF positional constraint enforcing that the pointing line through the finger reaches a desired target, while precisely pinpointing a button on a dial pad needs a 3–DOF positional constraint (the target for pointing), and an optional rotational constraint for determining a preferred pointing orientation style (see Figure 4).

The optimization starts by selecting k example motions M_j from the example motion cluster that best satisfy the constraint function f, $j = \{1, \ldots, k\}$. For example, in a typical reaching task, the k motion examples having the hand joint closest to the target will be selected. For the case of reaching motions, the hand location at the final pose of the motion is typically used as the parameterization frame. For gestures, the frame of the gesture stroke point is used.

Our optimization procedure is based on a traditional but efficient motion blending scheme, where an initial blended motion $\boldsymbol{M}$ is obtained with $\boldsymbol{M}(\boldsymbol{w}) = \sum_{j=1}^{k} w_j M_j$, where $\boldsymbol{w} = \{w_1, \ldots, w_k\}$ are blending weights initialized from a traditional RBF interpolation scheme. Any suitable kernel function can be used and we employ the popular $exp^{-\|\boldsymbol{e}\|^2/\sigma^2}$ kernel. Since our optimization runs on-line during interaction with the user, we do not attempt to optimize kernel functions in respect to the constraints [18, 22]. Instead, our blending weights will be optimized independently of the interpolation kernel.

In order to enforce a given constraint C, our goal is to find the optimal set of blending weights $\boldsymbol{w}$, which produces the minimum error e^*, measured by the constraint error function f:

$$e^* = min_{w_j \in [0,1]} \; f\left(\sum_{j=1}^{k} w_j M_j\right). \tag{2}$$

This formulation can also account for multiple constraints by combining the error metric of each constraint in a single weighted summation. Two coefficients are then introduced for each constraint C_i, $i = \{1, \ldots, n\}$: a normalization coefficient n_i and a prioritization coefficient c_i. The purpose of coefficient n_i is to balance the magnitude of the different error metrics associated to each constraint. Coefficient c_i allows the specification of relative priorities between the constraints.

The result is essentially a multi-objective optimization problem, with the goal being to minimize an error metric composed of the weighted summation of the individual error metrics:

$$e = \; min_{w_j \in [0,1]} \sum_{i=1}^{n} \left(c_i \; n_i \; f_i \left(\boldsymbol{M}(\boldsymbol{w})\right)\right). \tag{3}$$

Independent of the number of constraints being addressed, when constraints are fully satisfied, $e \rightarrow 0$. Figure 4 shows several results obtained by our optimization scheme.

Several optimization routines were implemented for solving our inverse blending problems, including: steepest ascent hill-climbing, the Nelder-Mead method and the gradient decent method [21]. Performance evaluations were conducted by solving 5000

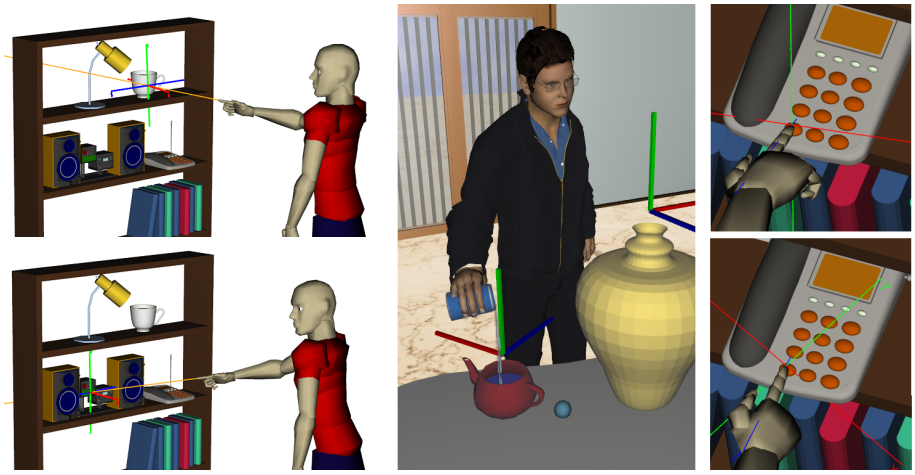


Fig. 4. The image shows results obtained with three different motion clusters. (a) Pointing motions parameterized by a 2–DOF directional constraint results in precise pointing to distant targets. (b) Pouring motions can be parameterized by a 2–DOF planar constraint specifying the precise location above the pouring target, and an additional constraint specifying an acceptable height range, so that liquids can correctly flow down into containers. (c) Precise pinpointing to given targets requires a 3–DOF positional constraint, with optional rotational constraints for further controlling the final poses obtained. The shown pinpointing examples show different orientations obtained, which match the x-axis of the tri-axes manipulator.

inverse blending problems for different scenarios: pointing, pouring and grasping. The Nelder-Mead method [21] has been proved to be the method of choice for our case where k remains below 15. The method requires a simple implementation and can typically achieve optimal blending weights within 2 milliseconds of computation time.

With suitable example motions in a given cluster, inverse blending can produce motions exactly satisfying given spatial constraints and fast enough for real-time applications. The several examples presented by this paper demonstrate its successful execution in different scenarios. To evaluate the performance of our method, a reaching task was designed to measure the errors produced by our method against a single RBF interpolation, with the 16 reaching motions in the database from Mukai and Kuriyama [18]. A total of 114 reaching goals (each specifying a 3–DOF positional constraint) were placed evenly on a spherical surface within reach of the character. These goals are highlighted with small yellow dots in Figure 5. The end locations of the hand trajectory in each example motion are shown as gray dots.

For each reaching target on the surfaces shown in Figure 5, we first used the RBF interpolation alone to generate a reaching motion and recorded the final hand position where the character actually reaches. These final positions are used to construct a mesh grid, which is shown on the upper row of Figure 5. Each triangle on the mesh is colored in respect to the average errors from its vertices, representing the distance error between the final hand positions and their corresponding reaching targets. We then use inverse blending optimization to perform the same tasks, and the mesh constructed is shown

on the lower row of Figure 5. The reaching motions generated by inverse blending can precisely reach most of the targets, and the measured errors were practically zero across most of the mesh. Only at the boundary of the surface that errors start to appear. In this specific task, the radius of the spherical surface was set to $80cm$, and both methods used eight example motions from the database ($k = 8$) for computing each reaching task.

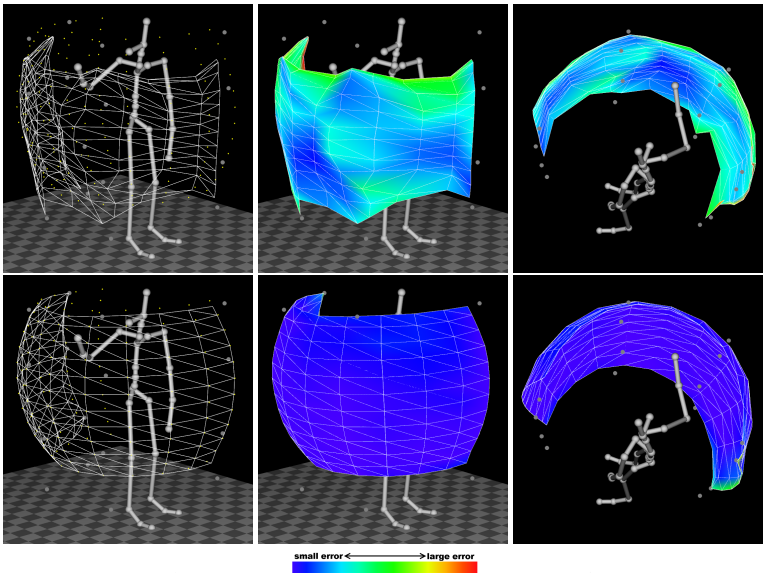


Fig. 5. Visualization of errors obtained by RBF interpolation and inverse blending. The upper row shows the results obtained with RBF interpolation. The blue smooth meshes on the lower row show the inverse blending results, which can well satisfy the given 3–DOF positional constraints.

It is important to note that the ability of enforcing constraints greatly depends on the existing variations among the used motion examples being blended. The number of needed example motions also depend on the size of the target volume space. The computational time required for finding solutions will also depend on the quality and number of considered motion examples (the k value). However, as showed in our several examples, these limitations can be well addressed by appropriately modeling example motions, and balancing the coverage vs. efficiency trade-off specifically for each action being modeled.

6 Results and Discussion

Examples demonstrating several aspects of our interactive system are presented in the video accompanying this paper (available at `http://graphics.ucmerced.edu/`).

We believe that our system achieves an effective overall design for modeling parameterized motions, which are extremely important for several types of applications (see

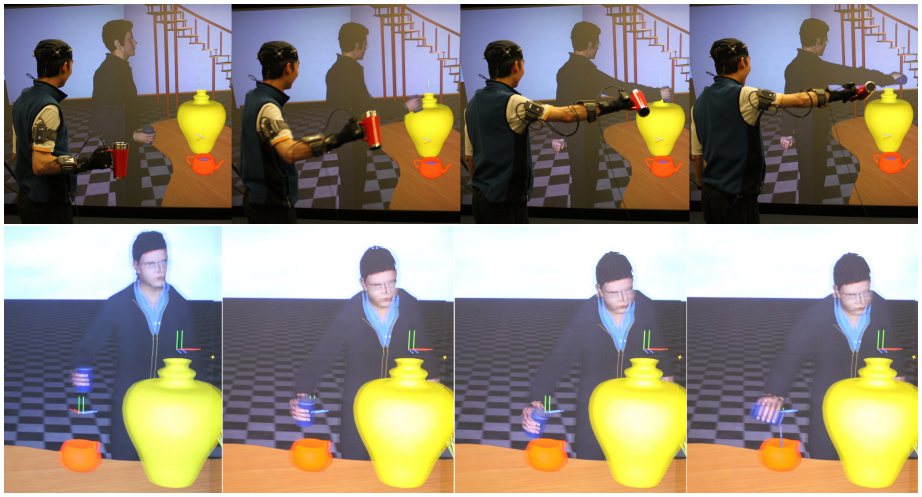


Fig. 6. In this example pouring motions are demonstrated and added to a parameterized pouring action cluster (top sequence). Once completed, the cluster is used to generate new pouring actions to arbitrary locations (bottom sequence).

Figure 6). Our framework well addresses the modeling of generic gestures and actions to be executed by interactive virtual humans, and furthermore allows non–skilled animators to intuitively obtain realistic results. Our proposed motion interface has therefore the potential to impact many applications.

Our first results open several new opportunities for further development of our interface design. For instance, we intend to quantify the advantages and limitations of employing an immersive stereo visualization display, and further explore the benefits of displaying the user's avatar during the modeling phase, as opposed to simply rely on an immersive direct motion performance. Direct performance offers better precision when interacting with the virtual environment, however note that the user is not required to demonstrate motions exactly meeting constraints, since the inverse blending will be responsible for that during the training phase.

Several improvements to our current motion capture interface can also be performed. For instance, a two-arm tracking interface can be easily integrated by including a few voice commands and/or finger gestures for operating the system, in order to free the left hand from holding the WiiMote controller. The integration of locomotion is also important, and in the scope of our target applications, we plan to include automatic locomotion and body positioning algorithms for controlling the virtual human instead of learning lower-body locomotion animations from the user. An independent gaze model also appears to be necessary.

We also intend to apply our system to concrete training scenarios. Additional constraints and parameterizations will be included, for instance related to parameterizing motion variations according to emotional levels, importance, etc. Our inverse blending

technique can handle any generic parameterizations as long as suitable motion clusters can be provided.

7 Conclusions

We have presented in this paper a novel motion modeling framework based on the direct demonstration and parameterization of motions. We have in particular presented the several algorithmic solutions required for enabling the development of the proposed design: a fast procedure for motion mapping from a reduced marker set, an intuitive motion interface for enabling the direct demonstration of parameterized actions and gestures, and an inverse blending optimization technique able to efficiently achieve realistic and parameterized motions in respect to arbitrarily given targets.

Our proposed framework has been implemented in an immersive multi-tile stereo visualization system, achieving a powerful and intuitive interface for programming generic parameterized motions by demonstration. We believe that the overall concept of our system has the potential to impact many applications.

Acknowledgements This work was partially supported by NSF Awards IIS-0915665 and CNS-0723281, and by a CITRIS seed funding grant.

References

1. Arikan, O., Forsyth, D.A., O'Brien, J.F.: Motion synthesis from annotations. ACM Transaction on Graphics (Proceedings of SIGGRAPH) 22(3), 402–408 (2003)
2. Billard, A., Matarić, M.J.: Learning human arm movements by imitation: Evaluation of a biologically inspired connectionist architecture. Robotics and Autonomous Systems 37:2-3, 145–160 (November, 30 2001)
3. Bruderlin, A., Williams, L.: Motion signal processing. In: SIGGRAPH '95. pp. 97–104. ACM, New York, NY, USA (1995)
4. Chai, J., Hodgins, J.K.: Performance animation from low-dimensional control signals. In: SIGGRAPH '05: ACM SIGGRAPH 2005 Papers. pp. 686–696. ACM, New York, NY, USA (2005)
5. Cooper, S., Hertzmann, A., Popović, Z.: Active learning for real-time motion controllers. ACM Transactions on Graphics (SIGGRAPH 2007) 26(3) (Aug 2007)
6. Dontcheva, M., Yngve, G., Popović, Z.: Layered acting for character animation. ACM Transactions on Graphics 22(3), 409–416 (2003)
7. D.Raunhardt, R.: Motion constraint. Visual Computer pp. 509–518 (2009)
8. Gebhard, P., Kipp, M., Klesen, M., Rist, T.: What are they going to talk about? towards life-like characters that reflect on interactions with users. In: Proc. of the 1st International Conference on Technologies for Interactive Digital Storytelling and Entertainment (TIDSE'03) (2003)
9. Grochow, K., Martin, S., Hertzmann, A., Popović, Z.: Style-based inverse kinematics. ACM Transactions on Graphics (Proceedings of SIGGRAPH) 23(3), 522–531 (2004)
10. Huang, Y., Kallmann, M.: Interactive demonstration of pointing gestures for virtual trainers. In: Proceedings of 13th International Conference on Human-Computer Interaction. San Diego, CA (2009)
11. J. Yamaguchi, A. Takanishi, I.K.: Development of a biped walking robot compensating for three-axis moment by trunk motion. Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems pp. 561–566 (1993)

12. Kallmann, M.: Analytical inverse kinematics with body posture control. Computer Animation and Virtual Worlds 19(2), 79–91 (2008)
13. Kopp, S., Wachsmuth, I.: Model-based animation of co-verbal gesture. Computer Animation, 2002. Proceedings of pp. 252–257 (2002)
14. Kovar, L., Gleicher, M.: Automated extraction and parameterization of motions in large data sets. ACM Transaction on Graphics (Proceedings of SIGGRAPH) 23(3), 559–568 (2004)
15. Kovar, L., Gleicher, M., Pighin, F.: Motion graphs. ACM Trans. Graph. 21(3), 473–482 (2002)
16. Lee, J., Chai, J., Reitsma, P.S.A., Hodgins, J.K., Pollard, N.S.: Interactive control of avatars animated with human motion data. ACM Transactions on Graphics 21(3), 491–500 (2002)
17. Monheit, G., Badler, N.I.: A kinematic model of the human spine and torso. IEEE Comput. Graph. Appl. 11(2), 29–38 (1991)
18. Mukai, T., Kuriyama, S.: Geostatistical motion interpolation. In: ACM SIGGRAPH. pp. 1062–1070. ACM, New York, NY, USA (2005)
19. Noma, T., Zhao, L., Badler, N.I.: Design of a virtual human presenter. IEEE Computer Graphics and Applications 20(4), 79–85 (2000)
20. Peinado, M., Meziat, D., Maupu, D., Raunhardt, D., Thalmann, D., Boulic, R.: Full-Body Avatar Control with Environment Awareness. IEEE Computer Graphics And Applications 29, 62–75 (2009)
21. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical Recipes 3rd Edition: The Art of Scientific Computing. Cambridge University Press, New York, NY, USA (2007)
22. Rose, C., Bodenheimer, B., Cohen, M.F.: Verbs and adverbs: Multidimensional motion interpolation. IEEE Computer Graphics and Applications 18, 32–40 (1998)
23. Rose, C.F., Sloan, P.P.J., Cohen, M.F.: Artist-directed inverse-kinematics using radial basis function interpolation. Computer Graphics Forum (Proceedings of Eurographics) 20(3), 239–250 (September 2001)
24. S. Kagami, e.a.: Autobalancer: An online dynamic balance compensation scheme for humanoid robots. Int. Workshop Alg. Found. Robot (2000)
25. Safonova, A., Hodgins, J.K.: Construction and optimal search of interpolated motion graphs. In: SIGGRAPH '07: ACM SIGGRAPH 2007 papers. p. 106. ACM, New York, NY, USA (2007)
26. Schaal, S., Ijspeert, A., Billard, A.: Computational approaches to motor learning by imitation. The Neuroscience of Social Interaction 1431, 199–218 (2003)
27. Stone, M., DeCarlo, D., Oh, I., Rodriguez, C., Stere, A., Lees, A., Bregler, C.: Speaking with hands: creating animated conversational characters from recordings of human performance. ACM Transactions on Graphics 23(3), 506–513 (2004)
28. Thiebaux, M., Marshall, A., Marsella, S., Kallmann, M.: Smartbody: Behavior realization for embodied conversational agents. In: Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS) (2008)
29. Unuma, M., Anjyo, K., Takeuchi, R.: Fourier principles for emotion-based human figure animation. In: SIGGRAPH '95. pp. 91–96. ACM, New York, NY, USA (1995)
30. Wiley, D.J., Hahn, J.K.: Interpolation synthesis of articulated figure motion. IEEE Computer Graphics and Applications 17(6), 39–45 (1997)
31. Y. Zheng, e.a.: Generating human interactive behaviours using the windowed viterbi algorithm. Computer Vision and Computer Graphics. Theory and Applications pp. 70–82 (2009)
32. Yamane, K., Kuffner, J.J., Hodgins, J.K.: Synthesizing animations of human manipulation tasks. In: SIGGRAPH '04: ACM SIGGRAPH 2004 Papers. pp. 532–539. ACM, New York, NY, USA (2004)