Planning Motions and Placements for Virtual Demonstrators

Yazhou Huang and Marcelo Kallmann

Abstract—In order to deliver information effectively, virtual human demonstrators must be able to address complex spatial constraints and at the same time replicate motion coordination patterns observed in human-human interactions. We introduce in this paper a whole-body motion planning and synthesis framework that coordinates locomotion, body positioning, action execution and gaze behavior for generic demonstration tasks among obstacles. Human-like solutions are achieved with a coordination model extracted from experiments with human subjects. Given an observer location and a target demonstration to be performed, the proposed planner automatically identifies body placements respecting visibility constraints, locomotion accessibility, and action feasibility among obstacles. Actions are modeled with clusters of example motions and a fast collision avoidance procedure in blending space is introduced to avoid nearby obstacles when needed. Locomotion towards new placements integrates planning among obstacles and is based on a motion capture database organized for efficient synthesis of motions with precise path following and arrival constraints. The proposed solution introduces effective approaches for modeling and solving complex demonstrative tasks for interactive applications.

Index Terms—Virtual Trainers, Motion Planning, Intelligent Virtual Humans.

1 Introduction

Virtual humans and embodied conversational agents are promising in the realm of human-computer interaction applications. One central goal in the area is to achieve virtual assistants that can effectively interact, train, and assist people in a wide variety of tasks. The need to demonstrate objects and procedures appears in many situations; however, the underlying motion synthesis problem is complex and has not been specifically addressed before. Simple everyday demonstrations involve a series of coordinated steps that a virtual agent needs to replicate. The agent needs to walk while avoiding obstacles along the way, stop at an appropriate demonstration location with clear view to the target and observer, interact with the object (e.g. point to it and deliver information), and also maintain visual engagement with the observer. This work addresses such harmonious multi-level orchestration of actions and behaviors (see Figure 1).

The proposed model was built from experiments with human subjects where participants were asked to freely approach target objects at different positions and to deliver object information to observers at various locations. These experiments provided ground truth data for defining a coordination model that is able to orchestrate the involved pieces of a demonstration task. The result is a whole-body motion planning framework, called PLACE, that addresses the five main pieces of the problem in an unified way:

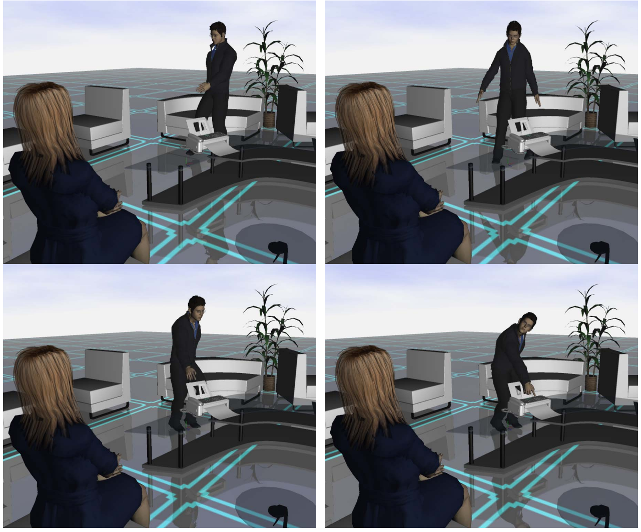

Fig. 1. Our PLACE planner synthesizes whole-body demonstrations for arbitrarily located targets and observers, also taking into account obstacles and visual occluders.

- Placement: optimal character placement is essential for addressing target and observer visibility, locomotion accessibility, and action execution constraints;
- Locomotion: locomotion synthesis among obstacles and towards precise placements allows the character to position itself in order to perform a demonstration;
- Action: realistic action execution needs to address arbitrary object locations and to avoid nearby obstacles when needed;
- Coordination: coordination is important for transitioning well from locomotion to the upper-body demonstrative

- *Y. Huang performed this work while at the University of California, Merced. He is now with EON Reality, Inc. E-mail: yhuang6@ucmerced.edu*
- *M. Kallmann is with the University of California, Merced. E-mail: mkallmann@ucmerced.edu*

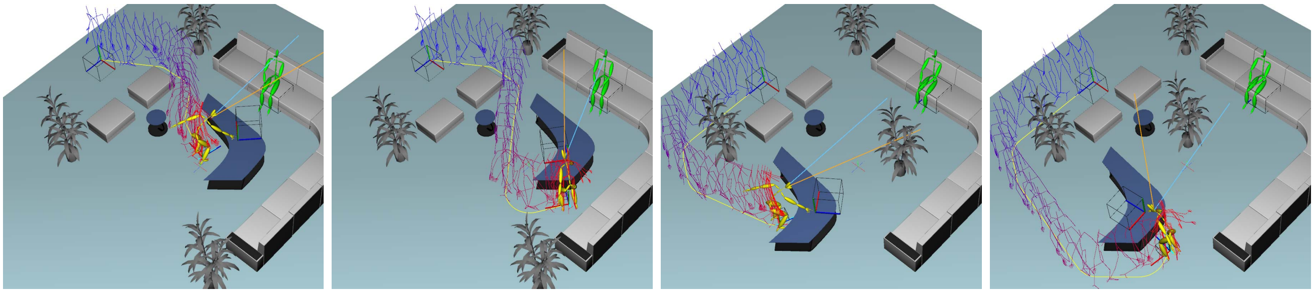

Fig. 2. From left to right: in the first two scenarios the computed demonstrations reasonably face the observer, while in the last two cases a visual occluder (the house plant) leads to solutions with non-trivial placements. The orange and blue lines respectively represent the head and the eye gaze orientations, at the demonstration action stroke point. The resulting gaze always reaches eye contact with the observer.

action; and

- Engagement: observer engagement is obtained with a gaze behavior that interleaves attention to the observer and the target in order to achieve effective demonstrations.

The realism of the solutions is addressed at two levels. At the behavioral level, placement, coordination and engagement are solved following models extracted from experiments with human subjects. At the motion synthesis level, locomotion and actions are synthesized from collections of motion capture clips organized for efficient synthesis and coordination. The presented techniques were developed such that solutions can be computed at interactive rates in realistic, reasonably complex, environments. See Figure 2 for examples.

This paper is an extended version of our previous work [1]. In this extended version we present the employed locomotion synthesis module, which has been developed specifically for this work, and which represents an effective approach to address departures and arrivals with arbitrary body orientations. The locomotion module is presented in Section 5.

The main contribution of the overall work is the definition, modeling and effective solution of whole-body demonstrative tasks. The proposed techniques are the first to address the overall problem in an integrated fashion.

2 Related Work

The proposed behavioral model coordinates gesture with locomotion synthesis in order to synthesize natural-looking motions for demonstrative tasks. The related work analysis that follows is divided in the three main areas related to our work: behavioral models, gesture synthesis, and locomotion synthesis.

2.1 Behavioral Models

One central aspect of our work is a new positioning model for performing demonstrations. Perhaps the first general investigation of positioning behaviors in humans was performed by Scheflen and Ashcraft [2], who present a pioneering work introducing the concept of *territoriality* in human-human interactions. The work however does not address computational models.

More in general, life-like agents providing information to users have already been proposed in different contexts. André et al. [3] describe a life-like 2D interface agent which presents multimedia material to users combining planning with temporal reasoning to execute pointing gestures to the relevant parts of the material. Lester et al. [4] describe a testbed learning environment where a pedagogical agent moves in the environment and refers to objects in order to provide problem solving advice to users. A key difference in our work is that, instead of relying on simplified or cartoonish agents, we address the full motion synthesis problem of performing human-like demonstrations in complete 3D environments, and also addressing arbitrary positions for the observer.

Interactions between virtual walkers have also been investigated, with models derived from experimental data [5], however with focus on multi-agent collision avoidance behaviors. More recently, the concept of social territorial intelligence has been explored to control placement of agents interacting with each other [6] during multi-agent conversations. These works demonstrate the importance of placement models for virtual characters, however the specific problem of body positioning for performing demonstrative actions has not been addressed.

The fact that we consider demonstrations being performed with respect to an arbitrary observer also distinguishes our overall planning and motion synthesis problem from previous work. Addressing an observer is important for achieving realistic solutions and effective interactions with virtual humans.

For instance, it has been shown that visual engagement improves the amount of information memorized by an audience observing robotic storytellers [7] and narrative virtual agents [8]. Visual engagement is usually addressed by a suitable gaze behavior. Although previous work has focused on modeling gaze behavior in great detail [9], [10], little attention has been given to integration with full-body motion synthesis. In computer animation simple solutions have been employed [11], [12] based on predefined points of interest, however not associating with a complete set of events observed from human subjects during action execution and locomotion.

2.2 Gesture Synthesis

Several previous works have addressed computation models for synthesizing upper-body gestures and actions. While approaches based on procedural Inverse Kinematics solvers exist, the majority of previous works seek the realism achieved by methods based on motion capture data. Most of the methods rely on motion blending techniques by reusing motion capture examples for gesture synthesis.

A popular approach is to perform stroke-based blending of annotated gestures [12], often employing motion blending techniques suitable for parameterization and control [13], [14]. Our approach for action synthesis relies on available motion interpolation techniques [14], [15] but providing a new collision avoidance mechanism directly in the blending space, such that small adjustments in the gesture motion can accommodate for obstacles without loosing the overall quality of the motion. If the collision avoidance in not enough, the overall planner will seek for a more suitable body position to perform the action. This concept allows us to successfully address realistic scenarios with obstacles.

2.3 Locomotion Synthesis

With respect to data-based locomotion methods, several techniques have been proposed for achieving realistic and controllable locomotion synthesis [16]–[21]. Most of these methods have roots on the *motion graphs* approach [22], where different improvements are proposed to the general idea of transitioning between example clips in a given motion capture database. While several of these methods can probably be extended to address departures and arrivals with arbitrary position and orientation constraints, such an extension is not trivial.

Other approaches perform a variety of motion processing techniques in order to improve the controllability of the produced locomotion [23]. Our solution is based on this approach, however proposing a specific organization of locomotion clips that allows for fast locomotion synthesis ensuring departures and arrivals with arbitrary position and orientation constraints, and was mainly designed to ensure computation times suitable for interactive applications.

The locomotion planning problem becomes particularly challenging when it has to be coordinated with upper-body actions. Previous work [24], [25] has addressed the combination of arm planning (reaching or grasping) on top of locomotion, however the arm movements were basically superimposed onto walk sequences without a coordination model. Kim et al. [26] have addressed locomotion generation exhibiting varied upper-body expressive movements, however limited to the execution of expressive locomotion.

Additional methods have been developed for splicing upper-body actions from one motion to another [27]. For instance Heck et al. [28] proposed to splice upper-body actions from one motion to another preserving temporal and spatial alignment from full-body examples, van Basten and Egges [29] developed a splicing space without using full-body examples, however limited to base motions exhibiting a gait cycle, and more recently object manipulations have been coordinated with locomotion using physics controllers as a way to achieve plausible coordination [30]. These methods however have not addressed a coordination model for transitioning from locomotion into a demonstration action, a specific situation that happens in our present work and that involves different types of constraints.

Related methods based on sampling-based planners can also be observed. Multi-modal planning algorithms based on RRT- or PRM-like techniques have been proposed for coordinating multiple motion primitives [31]–[33] and could be employed for locomotion-action coordination. However these approaches lack the quality of motion capture synthesis, and such quality level is highly desired in computer animation. Other frameworks have incorporated data-driven synthesis involving some aspect of coordination [34]–[36], however not including the needed coordination to transition from body positioning towards viewer-dependent demonstration tasks.

In conclusion, the proposed approach addresses the new problem of whole-body demonstrations at multiple levels and uniquely integrates behavioral models from human subjects with realistic data-based motion synthesis.

3 Modeling Demonstrative Tasks

We have modeled the overall problem of synthesizing humanlike demonstrations with the help of experiments with human subjects. Our setup follows the approach in [37], but extending it for extracting complete motion models for demonstration tasks.

Four human participants were recruited to perform a variety of pointing tasks with full-body motion capture. Six small target objects T_i, $i \in \{1, \ldots, 6\}$, were placed on a horizontal coarse mesh grid and participants were asked to perform demonstration actions towards each T_i for a human observer O_j standing at five different positions around the targets ($j \in \{1, \ldots, 5\}$).

Each action consisted of pointing and delivering a short information about an object. Each configuration $\{T_i, O_j\}$ represented one trial per participant and generated one motion. A total of 30 distinct motions were generated per participant, each motion consisting of a complete pointing action with the associated locomotion and gaze behavior. The gaze typically moved several times between T_i and O_j. Each participant started from about 4 feet away from the mesh grid before walking towards the grid to point and describe T_i (see Figure 3). Participants therefore had to decide where to position themselves with respect to the targets and observers. The sequence of target selection was random and the targets were of similar size in order to reduce possible side effects related to their size [38].

The full-body capture data was annotated manually with an annotation tool specifically developed to mark and extract the parameters and timings of all relevant events in each trial. One of the most important behaviors observed was the chosen positioning that each participant used to perform the pointing action. The chosen position ensured

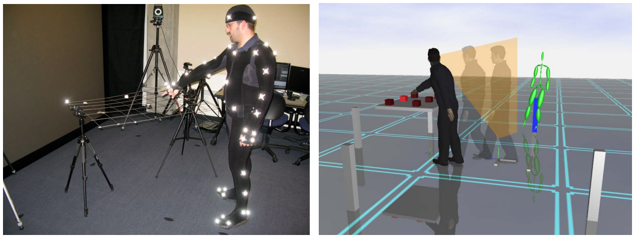

Fig. 3. Left: experiment setup. Right: illustration of one reconstructed motion. The observer location is represented with the green character and the maximum head orientation performed in the direction of the observer is shown with the orange plane.

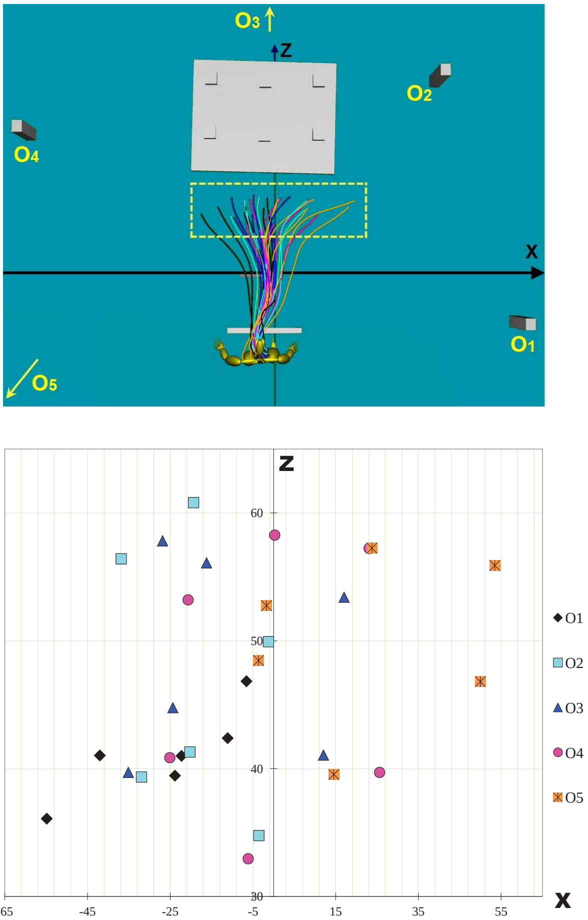

Fig. 4. Locomotion trajectories for each participant (top) and their ending positions in a closer look (bottom). The closer look region is identified with the yellow rectangle shown in the top picture. Motions with the same color are with respect to a same observer.

that the target and the observer were visible, and that the head rotation needed for eye contact with the observer was feasible. The position also ensured a successful execution of the action and with a fluid transition from locomotion. We now derive a generic placement model based on the observed data.

For each trial in the collected motion data we extracted the corresponding target position $\mathbf{p_t}$, the position of the observer $\mathbf{p_o}$, and the demonstrator position $\mathbf{p_d}$. Position $\mathbf{p_d}$ is the position used to perform the demonstration action, and is defined as the position when the locomotion is detected to completely stop, since there is a period when the action execution overlaps with the locomotion. Figure 4 plots locomotion trajectories and their corresponding final demonstration positions. The 5 distinct colors represent the 5 different observer locations. Each color appears 6 times, one for each target T_i. It can be observed that the demonstration positions do not show an obvious structure in global coordinates.

A local 2D coordinate system with origin at $\mathbf{p_t}$ is then used to derive our model. The coordinate system is illustrated with the XZ frame in Figure 5. The XZ frame can have arbitrary orientation, however it is more intuitive when the Z axis is orthogonal to the table border closest to the target. We can now use angles to locally encode all relevant placement parameters with respect to the target. The used local angles will not model the proximity of the demonstrator to the target, since this is a parameter that is action-dependent and we leave it as a free parameter in our model. For example a pointing motion can be executed with arbitrary distance to the target while this is not the case in a manipulation task. The angles considered by our model are the following (see Figure 5): the observer position α with respect to Z, the demonstrator position β with respect to -Z, the demonstrator's body orientation θ with respect to Z, and the maximum head rotation ϕ (in respect to Z) performed towards the observer.

The approach of expressing placement locally with respect to the action target correlates with the *axis concept* used for describing interaction connections [2]. By expressing the collected parameters with respect to our local coordinate system, the plots of their values nicely fit into clusters with good structure. This is shown in Figure 6. Since the proposed placement model shows good structure, we then performed nonlinear regressions in order to be able to estimate β, θ and ϕ as a function of an arbitrary input value for α. After smoothing the raw measurements with a least squares Savitzky-Golay filter, quadratic and cubic polynomial functions were fitted for β, θ and ϕ (see Appendix A for details).

The overall demonstration problem is then modeled as follows: given an upper-body demonstrative action A to be performed, the corresponding target object position $\mathbf{p_t}$, and the position of the observer $\mathbf{p_o}$, the goal of the PLACE planner is to synthesize a full-body motion for approaching the target and performing A with respect to $\mathbf{p_t}$ and for the observer located at $\mathbf{p_o}$. The planner solves the problem with the following steps. First, a suitable demonstration position $\mathbf{p_d}$ and body orientation $\mathbf{q_d}$ are determined by using the placement model and taking into account visual occluders, action feasibility and locomotion accessibility. Then, a locomotion sequence $L(\mathbf{p_d}, \mathbf{q_d})$ is synthesized for the character to walk from its current position to the demonstration placement $(\mathbf{p_d}, \mathbf{q_d})$. Action $A(\mathbf{p_t})$ is then

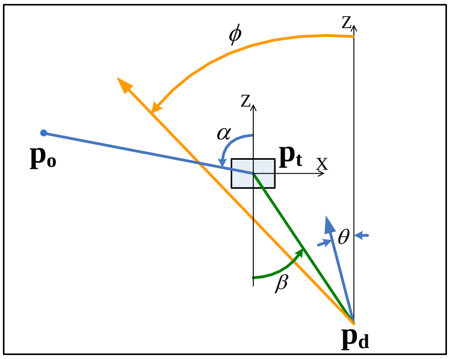

Fig. 5. Local coordinate system of the placement model. Angle α encodes the observer location, β the demonstrator location, θ the body orientation at action start, and ϕ encodes the maximum head rotation towards the observer.

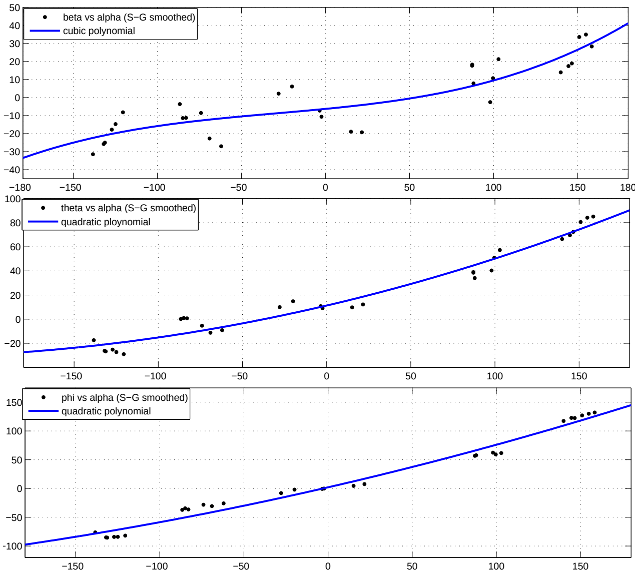

Fig. 6. Parameters of the placement model fitted with nonlinear regression on filtered data points. The horizontal axis is the α value and the vertical axis, from top to bottom, represents β, θ, and ϕ respectively. Values are in degrees.

synthesized and coordinated with the locomotion L. Finally, the head and eyes are animated to replicate the same gaze behavior patterns observed in the collected motions. These steps represent the five main components of PLACE and they are explained in the following sections.

4 Placement

Given the demonstrative action A, the target object position $\mathbf{p_t}$, and the observer position $\mathbf{p_o}$, the placement module will determine the optimal body position and orientation $(\mathbf{p_d}, \mathbf{q_d})$ for performing A.

First, the action synthesis module (described in Section 6) is queried for its preferred distance d_{pref} to execute $A(\mathbf{p_t})$. This distance denotes the preferred Euclidean distance between $\mathbf{p_t}$ and $\mathbf{p_d}$ so that the character will be more likely to succeed in performing the action. The computation of d_{pref} is action-dependent, it may be automatically selected according to reachability constraints if the demonstration has to achieve object contact, or in other cases (like in pointing), it can be a user-specified parameter fixed or dependent on features of the environment (like the size of the target).

The local reference frame of our placement model is then placed with origin at $\mathbf{p_t}$ and with the Z axis set to be orthogonal to the closest edge of the supporting table. At this point our local placement model can be applied in order to estimate a first candidate placement $(\mathbf{p_d^0}, \mathbf{q_d^0})$, where $\mathbf{p_d^0}$ is obtained by combining the estimated angle β with the preferred distance d_{pref}, and $\mathbf{q_d^0}$ represents the orientation estimated by θ in global coordinates. If $\mathbf{p_t}$ lies between $\mathbf{p_o}$ and $\mathbf{p_d^0}$, the Z axis of the local placement coordinate frame is re-oriented towards $\mathbf{p_o}$ and the initial placement $(\mathbf{p_d^0}, \mathbf{q_d^0})$ is re-computed. This will make the initial placement directly face the observer, a desired property in a placement.

Given a candidate placement, the placement is considered valid if: it does not lead to collisions with the environment, if $A(\mathbf{p_o})$ can be successfully executed from it, and if there is a collision-free path with enough clearance for the character to reach it. If a candidate placement is not valid due to a collision, it is tested again a few times with slightly perturbed body orientations and d_{pref} distances, thus increasing the chances of finding valid placements by local adjustment of the generated positions.

Several placements may be identified as valid and therefore we search for the optimal one with respect to visibility, head rotation comfort, and distance to the target.

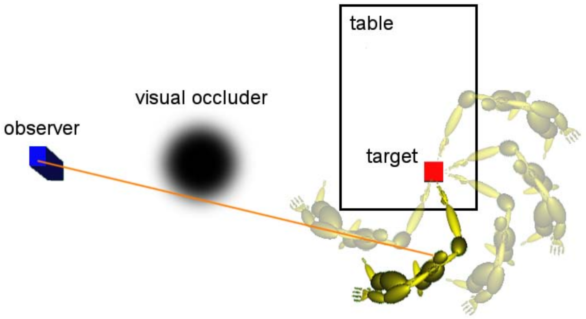

Fig. 7. Valid placements around the target are identified and ranked for selection. In the example, all shown postures are valid, except for the visibility criterion, which is satisfied by only one posture. The shown occluder with a fuzzy border illustrates that occluders may allow partially visibility, as is the case of the plant object shown in the scenario of Figure 2.

Starting from $(\mathbf{p_d^0}, \mathbf{q_d^0})$ we determine several valid placements $(\mathbf{p_d^k}, \mathbf{q_d^k})$ by adjusting the Z axis of the local model to new orientations around $\mathbf{p_t}$, for example by rotation increments of five degrees in both directions (see Figure 7). For each new orientation, the respective estimated placement is computed and tested for validity (adjusted if needed) and stored if valid. The search for valid placements may be set to be exhaustive with respect to the used rotation increment or to stop after a certain number of valid samples is found. The result of this phase is a set of K valid placements $(\mathbf{p}_\mathbf{d}^k, \mathbf{q}_\mathbf{d}^k)$, $k \in \{1, \ldots, K\}$. We then sort the placements in this set with respect to the following ranking cost function f_c:

$$f_c = e_{vis} * w_v + e_{neck} * w_n + e_{action} * w_a,$$

where e_{vis} is a measure of how occluded the observer is from the placement, e_{neck} is the amount of neck rotation required for reaching eye contact with the observer, e_{action} is the absolute difference between d_{pref} and the actual distance from $\mathbf{p}_\mathbf{d}^k$ to $\mathbf{p_t}$, and the scalar weights are constants used to adjust the relative contributions of each term.

The weights are adjusted such that the contribution of e_{vis} is significant, since uncomfortable (but feasible) placements are preferable to placements with bad visibility. The house plant in the scenarios of Figure 2 is an example of an object modeled with partial occlusion set to $e_{vis} = 50\%$. Candidate placements with bad visibility are discarded, as shown in Figure 7. Any other desired factor could be added to the ranking function as needed. For instance, the locomotion path length could be easily included in order to give preference to valid positions that are closer to the current position of the character.

The result is a sorted list of valid placements that can be used for executing A. The placement with minimum f_c is selected as the target demonstration location $(\mathbf{p_d}, \mathbf{q_d})$ to be used. Since this placement has been already checked for validity, it can be safely passed to the motion synthesis modules described in the next sections.

In case simplified (incomplete) validity tests are employed, the motion synthesis may happen to not be successful at some later stage, in which case the next best placement in the list can be used as the next alternative to be considered.

Figure 2 illustrates several body placements determined by our method, and Figure 8 exemplifies a typical high-ranked placement.

5 Locomotion Synthesis

The locomotion synthesis module needs to be able to address three main requirements:

- to be able to quickly check for locomotion accessibility to candidate placements when queried by the body placement module (Section 4), in order to allow for quick rejection of placements that offer no accessibility;
- to be able to synthesize motions that can navigate through narrow passages and with precise departure and arrival positions and orientations; and
- to produce purposeful motions resembling the ones observed in our experiments with human subjects, which consistently had sharp turnings (with small turning radius) at the departure and arrival locomotion phases.

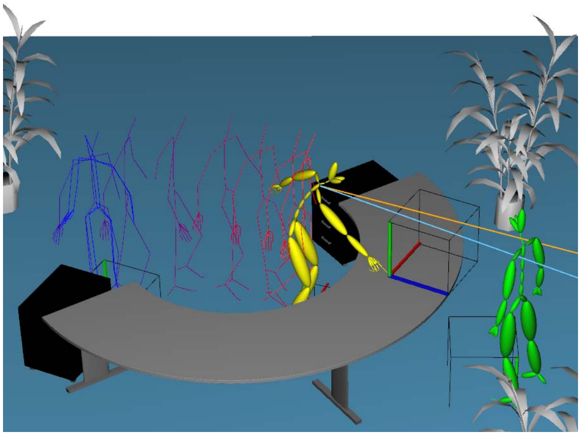

Fig. 8. The shown posture for the yellow character (the demonstrator) was selected for action execution by our algorithm. It exemplifies a typical high-ranked body placement with clear observer-demonstrator visibility, and direct frontal body orientation minimizing the amount of head rotation needed for achieving eye contact with the observer.

With respect to the first requirement, accessibility queries are computed with an efficient algorithm for computing paths [39], which is used to determine path feasibility with clearance under a few milliseconds of computation in our scenarios. With respect to the second and third requirements, we adopt a path following approach in order to be able to safely navigate through narrow passages of the environment. The problem is thus reduced to synthesizing a locomotion sequence that can follow the computed paths well, given that our computed paths already guarantee that there is enough clearance for the character to walk along them.

We have developed an optimized locomotion synthesis method based on a specific organization of locomotion clips from motion capture. Three specific types of locomotion sequences were collected with a full-body motion capture system: walk cycles, departure motions, and arrival motions.

Departure motions start from a rest posture and then walk towards different directions around the starting point using sharp-turn departures. Let M_d denote one departure motion, and $\mathcal{M}_\mathcal{D}$ denote the set of all departure motions collected. Similarly, an arrival motion M_a transitions from the walk cycle towards an arrival position and the set $\mathcal{M}_\mathcal{A}$ of all arrival motions will cover several possible arrival orientations.

In order to ensure coverage of arbitrary arrival and departure orientations, each motion is slightly deformed by accumulating and adding small orientation offsets at every frame. These offsets are applied over the root joint position and orientation. Let r and h represent two specified rotation offsets and let vector (x_i, z_i, θ_i) represent the trajectory of the root joint projected on the floor for a given clip, where: i is the clip frame index, x_i and z_i are the 2D projected coordinate offsets (current frame minus previous frame) on the floor, and θ_i is the yaw rotation of the character. In this formulation the first frame of the motion ($i = 0$) is skipped, therefore only values $i > 0$ are considered. Given r and h, we deform a clip by first applying, for every index i, a 2D rotation of r to vector (x_i, z_i) and by adding h to θ_i. The original root joint transformation is then updated for each frame such that its projected position reflects the new values of the deformed (x_i, z_i, θ_i) vectors.

As expected, whenever deformations are incorporated in the root joint, the original feet-floor constraints have to be recovered by employing standard Inverse Kinematics (IK) techniques. Since the deformation amounts are expected to be very small, we use an analytical IK solver applied to only the legs of the character in order to recover the constraints. If the amount of clips being used is high enough, we have observed that the needed corrections are often not noticeable.

The result of the described deformation procedure is that each motion can be "bent" to the left or to the right, and at the same time having the final body orientation of the character to be rotated a bit to the left or to the right, according to the given deformation amounts. In this way, each motion becomes parameterized to cover a small area around its original trajectory, and to arrive with a parameterized orientation at the final frame. As a result we obtain parameterized motions $M_d(r, h)$ and $M_a(r, h)$, where r controls the spatial coverage and h controls the final body orientation coverage. Zero values will result on the original motions.

Let $coverage(M)$ denote the "dilated region" covered by $M(r, h)$ for all accepted values of r and h, which are kept inside small bounds for ensuring that no noticeable artifacts are introduced to the motion. The coverage of a motion set $coverage(\mathcal{M})$ is the union of the coverage of all motions in the set. Figure 9 illustrates the concept of departure and arrival motions with their respective coverages. The character starts at the black arrow and its root joint follows the blue trajectories.

The used walk cycles contain variations between straight walk and sharp turns of different amounts. The collected motion clips were segmented and organized manually. We have collected 38 arrival motions and 31 departure motions, and we have then mirrored them with respect to the vertical sagittal plane of the character in order to further increase the coverage of the clips. Figure 10 illustrates the trajectories of the original arrival and departure clips that were used.

Given a path towards the selected placement, the locomotion synthesis is performed as follows. First, the departure and arrival clips $M_d(r, h)$ and $M_a(r, h)$ that best fit the path at its end points are selected and correctly placed best matching the exact start and goal body positioning and orientation.

Next, walk cycle clips M_c are repeatedly applied and concatenated to closely follow the path, with the same deformation strategies applied to M_c in order to ensure a close fit. At the end of this process, the final walk cycle will end nearby $M_a(r, h)$ but will most often not match closely enough for a simple blend-in. We then employ an inverse blending optimization procedure to generate one specific stepping motion M_{invb} so that the feet positions and body orientation best match the end of the incoming M_c and the beginning of $M_a(r, h)$, thus smoothly transitioning from the cyclic walk into the arrival clip.

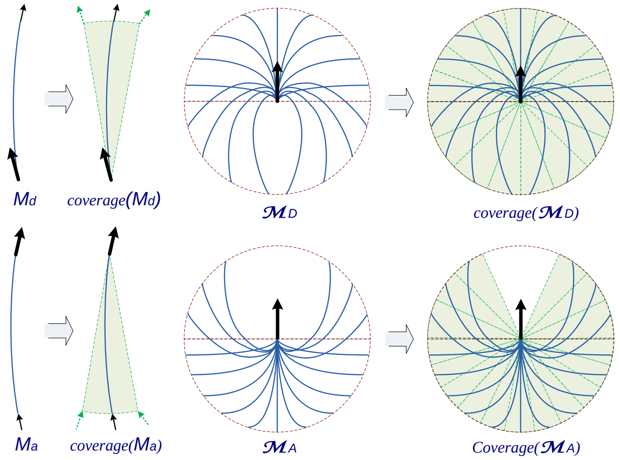

Fig. 9. Illustration of the departure (top row) and arrival (bottom row) motions used and the coverage of their deformation parameterization. Full coverage of departure and arrival configurations ensures precise arrival at arbitrary placement locations.

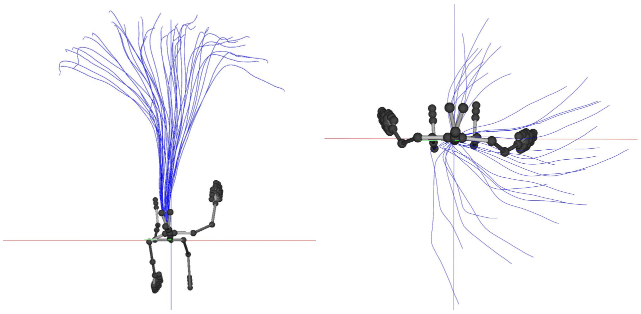

Fig. 10. Illustration of the original trajectories of the used arrival (left) and departure (right) motions. These original motions were mirrored with respect to the vertical sagittal plane of the character in order to cover arbitrary arrival and departure orientations.

The inverse blending procedure [15] will blend the available walk cycles with different blending weights per leg such that the legs achieve close proximity to allow a final transition. The inverse blending procedure is not needed when transitioning from M_d to M_c because there is no precise position to be achieved at the transition point, the only constraint being to closely align the final pose of M_d with the path trajectory.

Figures 11 and 12 illustrate the overall method. A complete sequence is typically computed within 50 milliseconds of computation time.

One specific case has however to be handled. Given that the departure and arrival clips take 2 to 3 steps to execute the involved sharp turnings, the overall procedure can only

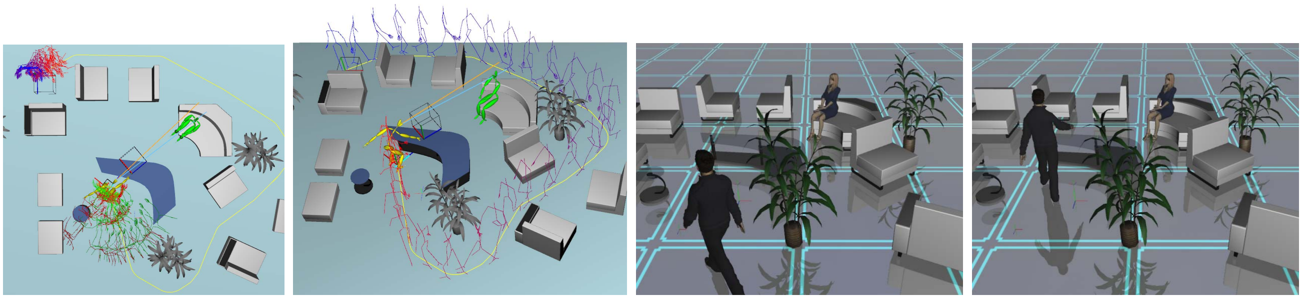

Fig. 12. Illustration of one particular locomotion sequence planned. From left to right: the departure and arrival clips nearby the path to be followed, skeleton trails illustrating the whole motion obtained, and two snapshots of the final result.

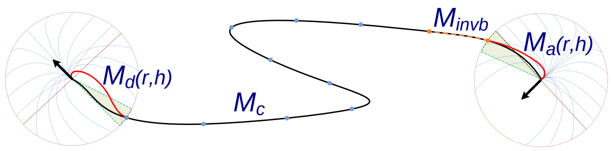

Fig. 11. A typical locomotion sequence will start with a departure motion, follow the path with multiple walk cycles, and then make a transition computed by inverse blending into the arrival motion.

handle sequences where the path length is long enough to accommodate at least one departure and one arrival clip. For short-range locomotion cases where the path is too short, the planner switches to only using special short-range clips M_s integrating departure and arrival movements, where each motion consists of 2 to 4 steps with deformation adjustments like in the regular case for improving coverage.

One example is shown in Figure 13. Coverage is however not easily guaranteed since the short-range stepping database has to cover many possible arrival and departure positions and orientations with a single dataset. The allowed bounds for the deformation parameterization also affects the obtained coverage. In most of the cases however, even if a precise body placement is not reached, the action execution can still be performed from a nearby location and the overall demonstration sequence can be successfully performed.

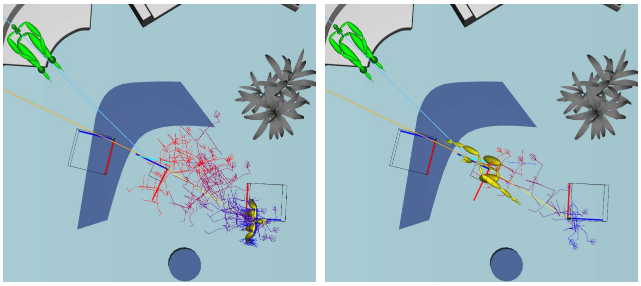

Fig. 13. Searching for short-range locomotion clips (left), and the achieved solution (right). Please refer to the first two images of Figure 12 for additional views of the same scenario.

6 Action Synthesis

The synthesis of the demonstrative action is performed with blending operations in a cluster of example motions. Given the target position $\mathbf{p_t}$ to be addressed by the end-effector at the stroke point of the action, blending weights are determined by inverse blending optimization [15] in order to address the target precisely. See Figure 14.

Collision avoidance has shown to be important. It not only increases the ability to find solutions in cluttered environments but it also improves the number of successful placements to be considered for action execution. We have developed a collision avoidance method that operates on the blending space of the example motions defining an action.

Blending space operations have been employed before in motion planning [36], but here we develop a faster collision avoidance procedure that does not require expensive planning around the obstacles. We create repulsive force fields in 3D and compute a scalar potential of collision P_c that encodes the potential of collision between the agent's end-effector E and the obstacles. Instead of computing the force field in discretized 2D cells [40], we approximate the bounding volume of the nearby obstacles with small spheres S_i (see Figure 14-top right), so that $P_c = exp(-\sum distance(E, S_i))$.

Let $\mathbf{p_t}$ be the target object to be addressed by action A. First, blending weights $\mathbf{w_t}$ that generate action $A(\mathbf{p_t})$ are computed by inverse blending. The produced motion can be re-written as a sequence of frames $F_i(\mathbf{w}_i)$, $i \in \{1, \dots, n\}$, and initialized with $\mathbf{w}_i = \mathbf{w_t}$, $\forall i \in \{1, \dots, n\}$. Next, we make a single pass from F_1 to F_n and adjust intermediate frames F_j at a given discretization resolution. The resolution is relative to the distance covered by the end-effector. Given a frame $F_j(\mathbf{w}_j)$ being visited, if its corresponding posture collides or is detected to be too close to an object, $\mathbf{w}_j$ is adjusted by inverse blending in order to minimize P_c at F_j, essentially shifting F_j away from the obstacles. Collisions are checked intermittently at mesh level and the process moves on to F_{j+1} when F_j becomes collision-free. Each time a frame is adjusted, the weights of the nearby frames (according to a smoothness window) are updated so that the overall sequence of weights is smooth, producing a new final motion $A(\mathbf{p_t})$ that smoothly avoids the nearby obstacles.

Figure 20 shows an example where the upper body action generates collisions, and the force field in blending space guides a deformed trajectory for the agent's end-effector so that the action can still be successfully completed with good clearance from the obstacles.

The method typically solves action synthesis under 300 milliseconds, with most computation time spent on mesh collision checking. The approach is able to control how much deformation is allowed, thus controlling the balance between action motion quality, action adaptation to obstacles, and body placement search time.

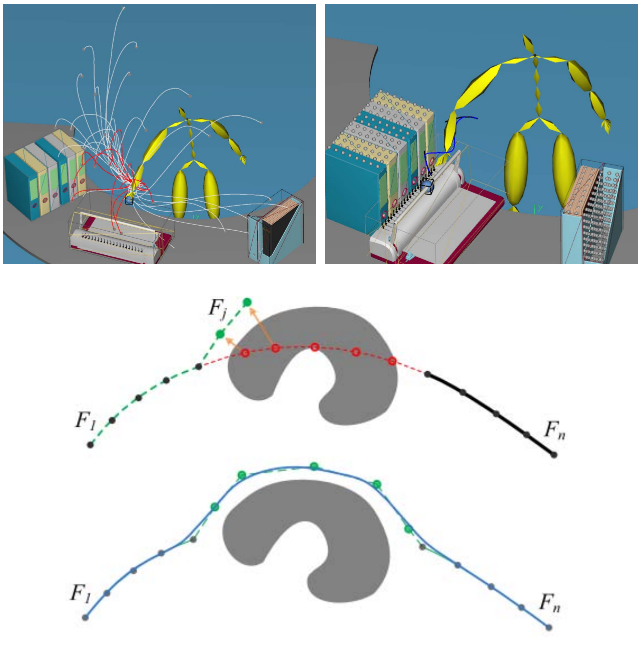

Fig. 14. Top-left: trajectories of one pointing database with 10 blended motions for one solution marked in red. Top-right: spheres are used to approximate the nearby objects and to compute P_c. Bottom: If intermediate frames collide their blending weights are adjusted to remove the collision.

7 Locomotion-Action Coordination

The locomotion transition into the action requires special attention in order to generate realistic results. We start by using a transition window of 0.58 seconds, which is the average window observed from our studies with human subjects. The window tells how early, before finishing the locomotion, the action should start to be executed. The action will start gradually taking control over the upper-body and will achieve full control when the locomotion stops. An important coordination problem that we address here is to ensure that the resulting arm swing pattern during the transition remains realistic.

Let S_l^{end} be the arm swing at the end of the locomotion sequence, and S_a be the arm swing direction of the action. In our examples the right arm is used by the action but the presented analysis can be equally employed to both arms. Two main cases may happen: 1) the arm swings can be codirectional, in which case a natural transition is automatically achieved, or 2) the arm swings can be contradirectional, what would cause a sudden change in the arm swing direction during the transition window.

Sudden changes in the arm swing were never observed in our experiments with human subjects, who were very good at achieving coherent final steps with clear final arm swings. We therefore fix contradirectional cases in two possible ways. If the final locomotion swing S_l^{end} slightly overlaps into the action arm swing S_a, it is then shortened to match S_a and without having it to return to its target rest position. This is accomplished by repeating the final frames of S_l^{end}, skipping the same amount of initial frames of S_a, then smoothly blending into the latter. If however the final locomotion swing shows a significant overlap, S_l^{end} is then dropped and the previous swing cycle S_l^{prev} is extended to override S_l^{end}, before blending into S_a. We examine the swing velocities generated from both and the one showing better consistency is applied. Figure 15 illustrates the process.

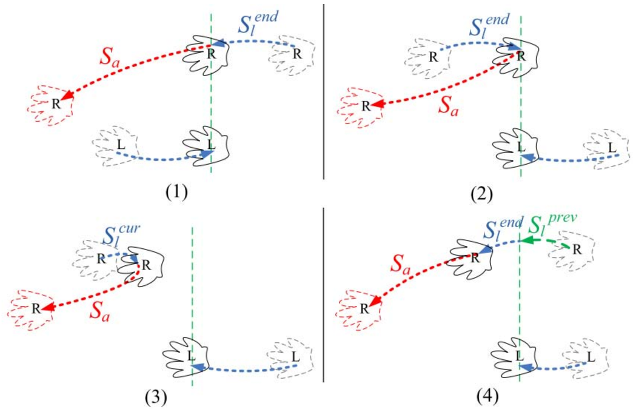

Fig. 15. Overlapping transition of the final arm swing of the locomotion S_l^{end} towards the arm swing direction generated by the action S_a. Codirectional cases can be directly blended (1), however contradirectional cases (2) have to be adjusted either by shortening the final locomotion swing (3) or by overriding it with the previous swing (4).

The overall transition blending is synthesized in two steps. First the action motion is temporally aligned with the end of the locomotion based on the arm swing patterns as explained before. If adjustments are made to the blending window, we make sure it remains between $1/3$ to $4/3$ seconds, following observations from user studies on blending artifacts and just-noticeable differences [41]. Then, after the locomotion has finished, Inverse Kinematics is applied to fix both feet on the floor while movements on the hips joint are allowed by the action in case the character needs to bend the knees to reach targets close to the floor, generating a final coherent whole-body action. Figure 16 illustrates the overall timeline of the coordination. As a result, the final transition appears smooth and human-like.

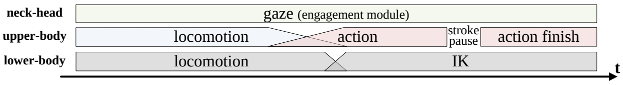

Fig. 16. Timeline of the locomotion-action coordination period.

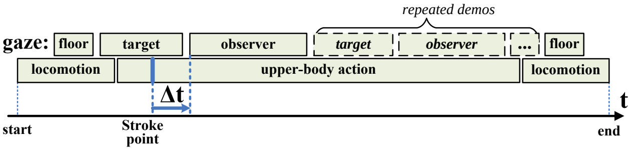

Fig. 17. Synchronization between gaze and motion events.

8 Engagement Module

The final step of PLACE includes a gaze model that follows the behavior observed in our experiments with human subjects.

We have statistically analyzed the motion data from the subjects in all the collected variations. We have observed that each demonstration trial consisted of a series of largely consistent gaze and gaze-related events where participants first gazed at the floor during the locomotion, and then gazed at the target and the observer during the upper-body action. Our gaze model generates gaze events that follow these observed patterns (see Figure 17).

We use a temporal delay Δt between the action stroke point and the start of the object gaze event that is correlated with the agent's head rotation angle ϕ. When the observer is not at the center of the agent's field of view, the gaze towards the observer starts before the action reaches the stroke point, resulting in a negative Δt.

This negative Δt was clearly observed in the collected data, and it can be explained by the fact that, when the observer is not inside the field of view of the demonstrator, the gaze needs to start earlier because more time is needed to rotate the head towards eye contact with the observer in time to be synchronized with the stroke point of the pointing motion.

The gaze behavior also incorporates gaze durations that decline over time across repeated demonstrations, an observed behavior in our experiments with human subjects. Figure 17 illustrates the main events of the gaze model. Additional observations about the gaze behavior are available in previous work [37].

9 Results and Discussion

Results are presented in Figures 1, 2, 18, 19 and 20. Additional results are presented in the accompanying video to this paper. The planner is capable of synthesizing entire sequences in a range from 100 to 400 milliseconds, depending on the complexity of the environment and the collision avoidance settings. Since it involves collision checking at the mesh level, the collision avoidance procedure is the most computational intensive module among all the modules.

Our results demonstrate that body placements are always chosen well and lead to positions clearly addressing all involved constraints. The coordination of the swing arm trajectories has also shown to produce good results.

Although our placement model was derived from real observers in standing postures, in the presented simulations the virtual scenes consider more natural situations with the virtual observers sitting in a sofa. We believe that this variation should not affect the model in any important way.

Example of motion capture sequences used in our blending procedures are available from the following project website: http://graphics.ucmerced.edu/software/invbld/. This website is being extended to include software and motion capture data from upper-body actions collected during the development of this project. The data will be made publicly available. The available data is organized in clusters and is ready for action execution with inverse blending (Section 6). An inverse blending algorithm implementation is also available.

The proposed methods can be extended in multiple ways. For instance the presented cost function to rank candidate placements currently does not account for path length. Total head rotation and path length are two practical measures that capture the effort required to perform a given placement and they can be incorporated as needed in the ranking function. We believe our overall model can be extended to moving targets by periodically querying the placement model and re-planning placement updates periodically.

In terms of limitations, our planner leaves out facial expressions and other behaviors that are specific to the context of the scenario being simulated. Context is one particular component that is important to be taken into account. For instance in emergency situations, shorter paths to placements would be preferable even if they lead to uncomfortable demonstrations with significant head rotation to achieve eye contact with the observer. Our model was also only designed to handle demonstrations for a single observer, although we believe that multiple observers can be easily incorporated if behavioral data exploring relevant possible configurations is obtained.

The key motivation behind this work is the goal to achieve natural "human-like" solutions. This paper demonstrates the importance of computing correct body placements in order to achieve this goal. A full action planner is not incorporated because it would produce solutions without quality guarantees. We therefore have only included a collision avoidance mechanism that accommodates for small collisions, in order to improve the solution space considered by the overall method. As a consequence, we cannot guarantee that a collision-free solution will always be found when one exists. Instead, our method only finds solutions that are deemed acceptable by the presented models and methods. As a result, only acceptable motions can be found by our method, and this is what makes sense in the context of information delivery. In this context, it is not reasonable to expect that a complex collision-free motion generated by a global planner would be necessary or even acceptable.

10 Conclusion

We have introduced a new behavioral and motor planning model for solving demonstrative tasks. Our proposed PLACE planner uniquely explores body placement tradeoffs involving visibility constraints, action feasibility, and locomotion accessibility. The proposed techniques can be computed at interactive rates and are suitable to several applications relying on interactive virtual humans as virtual trainers.

Acknowledgments This work was partially supported by NSF Award IIS-0915665.

Appendix A

The polynomials presented in this appendix fully define our placement model in local frame (see Figure 5). Angles are in degrees. The three polynomials describe the demonstrator position, orientation, and maximum head rotation with respect to the target position. The maximum head rotation limits the amount of head rotation allowed, and the remaining needed rotation for achieving eye contact with the observer is performed with eye rotation.

Polynomial (cubic and quadratic) functions were chosen over other types of fitting functions such as Gaussian and Fourier for better extrapolations when $\alpha > 150$ and $\alpha <$ -150. Details are given below.

- Demonstrator position:
$\beta = f(\alpha) = p_1\alpha^3 + p_2\alpha^2 + p_3\alpha + p_4$
Coefficients (with 95% confidence bounds):
$p_1 = 2.392\text{E}{-6}(-6.74\text{E}{-6}, 1.152\text{E}{-5})$,
$p_2 = 0.0003056(-0.0004444, 0.001056)$,
$p_3 = 0.1145(-0.04067, 0.2697)$,
$p_4 = -6.062(-15.42, 3.294)$.
Goodness of fit: $SSE = 5386$, $R^2 = 0.6156$,
$Adjusted\ R^2 = 0.5713$, $RMSE = 14.39$.

- Demonstrator orientation:
$\theta = f(\alpha) = p_1\alpha^2 + p_2\alpha + p_3$
Coefficients (with 95% confidence bounds):
$p_1 = 0.0006228(0.000262, 0.0009837)$,
$p_2 = 0.3267(0.2991, 0.3542)$,
$p_3 = 11.29(6.564, 16.02)$.
Goodness of fit: $SSE = 1441$, $R^2 = 0.9635$.
$Adjusted\ R^2 = 0.9608$, $RMSE = 7.304$.

- Maximum head rotation:
$\phi = f(\alpha) = p_1\alpha^2 + p_2\alpha + p_3$
Coefficients (with 95% confidence bounds):
$p1 = 0.0006673(0.0001145, 0.00122)$,
$p2 = 0.6736(0.6315, 0.7158)$,
$p3 = 2.073(-5.167, 9.312)$.
Goodness of fit: $SSE : 3381$, $R^2 : 0.9785$,
$Adjusted\ R^2 : 0.9769$, $RMSE : 11.19$.

References

[1] Y. Huang and M. Kallmann, “Planning motions for virtual demonstrators,” in *Proceedings of the 14th International Conference on Intelligent Virtual Agents (IVA)*, 2014.

[2] A. E. Scheflen and N. Ashcraft, *Human Territories: How We Behave in Space-Time*. Englewood Cliffs, NJ, USA: Prentice-Hall, 1976.

[3] E. André, T. Rist, and J. Müller, “Integrating reactive and scripted behaviors in a life-like presentation agent,” in *Proceedings of the Second International Conference on Autonomous Agents*, ser. AGENTS ’98. ACM, 1998, pp. 261–268.

[4] J. C. Lester, J. L. Voerman, S. G. Towns, and C. B. Callaway, “Deictic believability: Coordinating gesture, locomotion, and speech in lifelike pedagogical agents,” *Applied Artificial Intelligence*, vol. 13, pp. 383–414, 1999.

[5] J. Pettré, J. Ondřej, A.-H. Olivier, A. Cretual, and S. Donikian, “Experiment-based modeling, simulation and validation of interactions between virtual walkers,” in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA ’09. New York, NY, USA: ACM, 2009, pp. 189–198.

[6] C. Pedica and H. H. Vilhjálmsson, “Lifelike interactive characters with behavior trees for social territorial intelligence,” in *ACM SIGGRAPH 2012 Posters*, ser. SIGGRAPH ’12. New York, NY, USA: ACM, 2012, pp. 32:1–32:1.

[7] B. Mutlu, J. K. Hodgins, and J. Forlizzi, “A storytelling robot: Modeling and evaluation of human-like gaze behavior,” in *Proceedings of HUMANOIDS’06, 2006 IEEE-RAS International Conference on Humanoid Robots*. IEEE, December 2006.

[8] N. Bee, J. Wagner, E. André, T. Vogt, F. Charles, D. Pizzi, and M. Cavazza, “Discovering eye gaze behavior during human-agent conversation in an interactive storytelling application,” in *Int’l Conference on Multimodal Interfaces and Workshop on Machine Learning for Multimodal Interaction*, ser. ICMI-MLMI ’10. New York, NY, USA: ACM, 2010, pp. 9:1–9:8.

[9] K. E. Cullen, M. Huterer, D. A. Braidwood, and P. A. Sylvestre, “Time course of vestibuloocular reflex suppression during gaze shifts,” *Journal of Neurophysiology*, vol. 92, no. 6, pp. 3408–3422, 2004.

[10] M. R. Van Horn, P. A. Sylvestre, and K. E. Cullen, “The brain stem saccadic burst generator encodes gaze in three-dimensional space,” *J. of Neurophysiology*, vol. 99, no. 5, pp. 2602–2616, 2008.

[11] K. Yamane, J. J. Kuffner, and J. K. Hodgins, “Synthesizing animations of human manipulation tasks,” *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, vol. 23, no. 3, pp. 532–539, 2004.

[12] M. Thiebaux, A. Marshall, S. Marsella, and M. Kallmann, “Smartbody: Behavior realization for embodied conversational agents,” in *Seventh International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, 2008.

[13] T. Mukai and S. Kuriyama, “Geostatistical motion interpolation,” in *ACM SIGGRAPH*. New York, NY, USA: ACM, 2005, pp. 1062–1070.

[14] C. Rose, B. Bodenheimer, and M. F. Cohen, “Verbs and adverbs: Multidimensional motion interpolation,” *IEEE Computer Graphics and Applications*, vol. 18, pp. 32–40, 1998.

[15] Y. Huang and M. Kallmann, “Motion parameterization with inverse blending,” in *Proceedings of the Third International Conference on Motion In Games*. Berlin: Springer, 2010.

[16] R. Heck and M. Gleicher, “Parametric motion graphs,” in *Proceedings of the symposium on Interactive 3D Graphics and Games (I3D)*. New York, NY, USA: ACM Press, 2007, pp. 129–136.

[17] M. Gleicher, H. J. Shin, L. Kovar, and A. Jepsen, “Snap-together motion: assembling run-time animations,” in *Proceedings of the symposium on Interactive 3D Graphics and Games (I3D)*. New York, NY, USA: ACM Press, 2003, pp. 181–188.

[18] M. de Lasa, I. Mordatch, and A. Hertzmann, “Feature-Based Locomotion Controllers,” *ACM Transactions on Graphics*, vol. 29, no. 3, 2010.

[19] A. Treuille, Y. Lee, and Z. Popović, “Near-optimal character animation with continuous control,” in *Proceedings of ACM SIGGRAPH*. ACM Press, 2007.

[20] Y. Lee, K. Wampler, G. Bernstein, J. Popović, and Z. Popović, “Motion fields for interactive character locomotion,” *ACM Trans. Graph.*, vol. 29, no. 6, pp. 138:1–138:8, Dec. 2010.

[21] S. Levine, J. M. Wang, A. Haraux, Z. Popović, and V. Koltun, “Continuous character control with low-dimensional embeddings,” *ACM Trans. Graph.*, vol. 31, no. 4, pp. 28:1–28:10, Jul. 2012.

[22] L. Kovar, M. Gleicher, and F. Pighin, "Motion graphs," in *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*, ser. SIGGRAPH '02. New York, NY, USA: ACM, 2002, pp. 473–482.

[23] S. I. Park, H. J. Shin, T. H. Kim, and S. Y. Shin, "On-line motion blending for real-time locomotion generation," *Computer Animation and Virtual Worlds*, vol. 15, no. 3-4, pp. 125–138, 2004.

[24] C. Esteves, G. Arechavaleta, J. Pettré, and J.-P. Laumond, "Animation planning for virtual characters cooperation," *ACM Trans. Graph.*, vol. 25, no. 2, pp. 319–339, Apr. 2006.

[25] A. Shapiro, M. Kallmann, and P. Faloutsos, "Interactive motion correction and object manipulation," in *ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games (I3D)*, Seattle, 2007.

[26] Y. Kim and M. Neff, "Automating expressive locomotion generation," in *Transactions on Edutainment VII*. Springer, 2012, pp. 48–61.

[27] B. van Basten and A. Egges, "Motion transplantation techniques: A survey," *IEEE Computer Graphics and Applications*, vol. 32, no. 3, pp. 16–23, 2012.

[28] R. Heck, L. Kovar, and M. Gleicher, "Splicing upper-body actions with locomotion," in *Proceedings of Eurographics 2006*, september 2006.

[29] B. J. H. van Basten and A. Egges, "Flexible splicing of upper-body motion spaces on locomotion," *Computer Graphics Forum*, vol. 30, no. 7, pp. 1963–1971, 2011.

[30] Y. Bai, K. Siu, and C. K. Liu, "Synthesis of concurrent object manipulation tasks," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 156:1–156:9, Nov. 2012.

[31] K. Hauser and J.-C. Latombe, "Multi-modal motion planning in non-expansive spaces," *The International Journal of Robotics Research*, 2009.

[32] M. Kallmann, Y. Huang, and R. Backman, "A skill-based motion planning framework for humanoids," in *Proceedings of the International Conference on Robotics and Automation (ICRA)*, 2010.

[33] K. Hauser and V. Ng-Thow-Hing, "Randomized multi-modal motion planning for a humanoid robot manipulation task," *The International Journal of Robotics Research*, 2010.

[34] A. Feng, Y. Xu, and A. Shapiro, "An Example-Based Motion Synthesis Technique for Locomotion and Object Manipulation," in *Proceedings of the ACM Symposium on Interactive 3D Graphics and Games (I3D)*, 2012.

[35] J. Pan, L. Zhang, M. C. Lin, and D. Manocha, "A hybrid approach for simulating human motion in constrained environments," *Comput. Animat. Virtual Worlds*, vol. 21, no. 34, pp. 137–149, May 2010.

[36] Y. Huang, M. Mahmudi, and M. Kallmann, "Planning humanlike actions in blending spaces," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[37] Y. Huang, M. Kallmann, J. L. Matthews, and T. Matlock, "Modeling gaze behavior for virtual demonstrators," in *Proceedings of the 11th International Conference on Intelligent Virtual Agents (IVA)*, 2011.

[38] R. W. Soukoreff and I. S. MacKenzie, "Towards a standard for pointing device evaluation: Perspectives on 27 years of fitts' law research in hci," *International Journal of Human-Computer Studies*, vol. 61, pp. 751–789, 2004.

[39] M. Kallmann, "Dynamic and robust local clearance triangulations," *ACM Transactions on Graphics (presented at SIGGRAPH)*, vol. 33, no. 5, 2014.

[40] L. Zhang, S. M. LaValle, and D. Manocha, "Global vector field computation for feedback motion planning," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 3065–3070.

[41] J. Wang and B. Bodenheimer, "Computing the duration of motion transitions: an empirical approach," in *Proceedings of the 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, ser. SCA '04. Eurographics Association, 2004, pp. 335–344.

Yazhour Huang Yazhou Huang received his PhD from the University of California Merced in 2012, on the topic of motion capture based character animation for interactive virtual humans. During his time as a graduate student he worked as a graduate research assistant in the UC Merced Computer Graphics Lab. His research interests include motion parametrization, motion planning, virtual reality, human-computer interaction and robotics. Currently he is lead R&D engineer at EON Reality, Inc.

Marcelo Kallmann Marcelo Kallmann is founding faculty and associate professor of computer science at the University of California, Merced. He holds a PhD from the Swiss Federal Institute of Technology in Lausanne (EPFL), and was research faculty at the University of Southern California (USC) before moving to UC Merced in 2005. His areas of research include computer animation, virtual reality and motion planning. In 2012 he was program co-chair for the 5th International Conference on Motion in Games (MIG), in 2013 he was a guest co-editor of the Computer Animation and Virtual Worlds Journal (CAVW), and in 2014 he is program co-chair for SBGames and Associated Editor for ICRA 2015. His research has been funded by several NSF awards and his recent work on triangulations for path planning runs inside the recently announced The Sims 4.

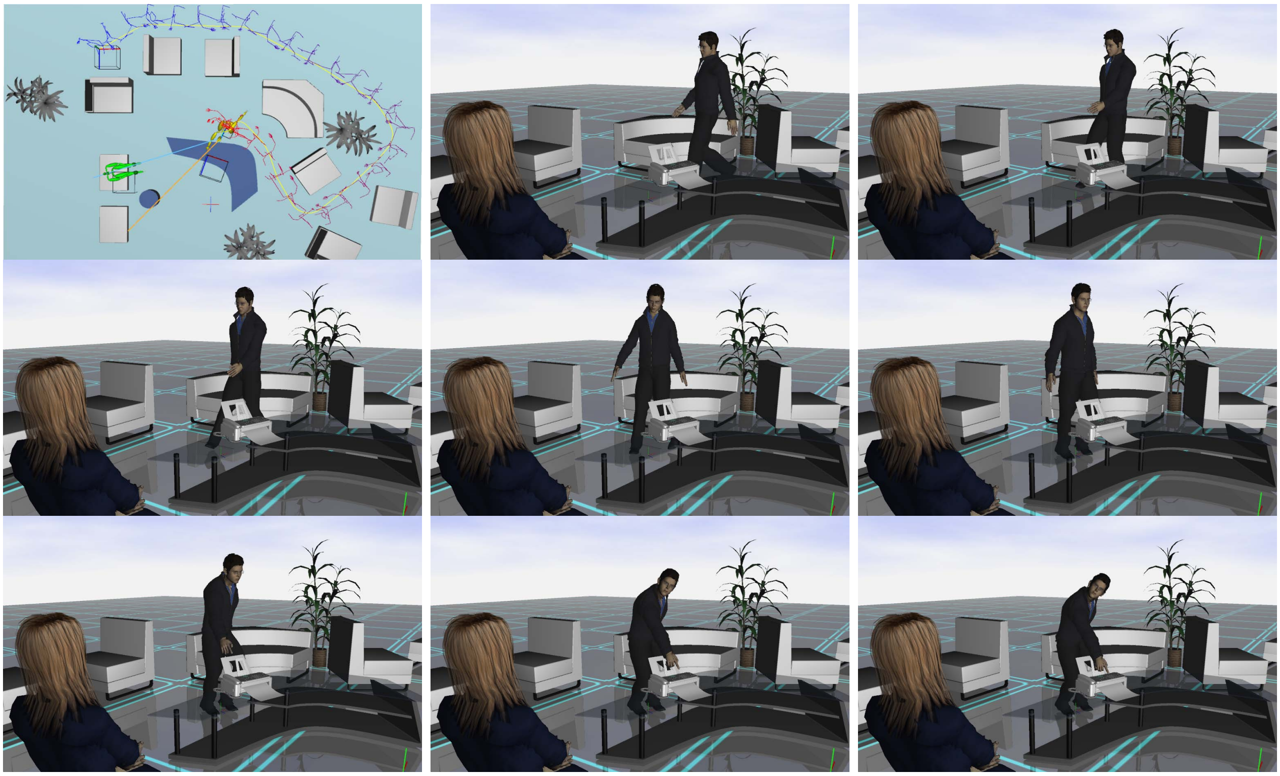

Fig. 18. Example of a solution produced by PLACE. The top-left image shows the planning scenario and the solution placement for execution of the demonstration. The following sequence of snapshots shows the arrival locomotion seamlessly transitioning into the demonstration action pointing at the fax machine with coordinated gaze towards the observer.

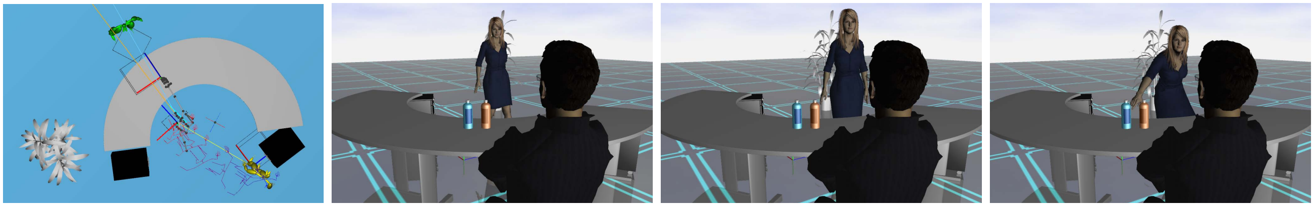

Fig. 19. Short-range solution suitable for pointing and describing the blue bottle.

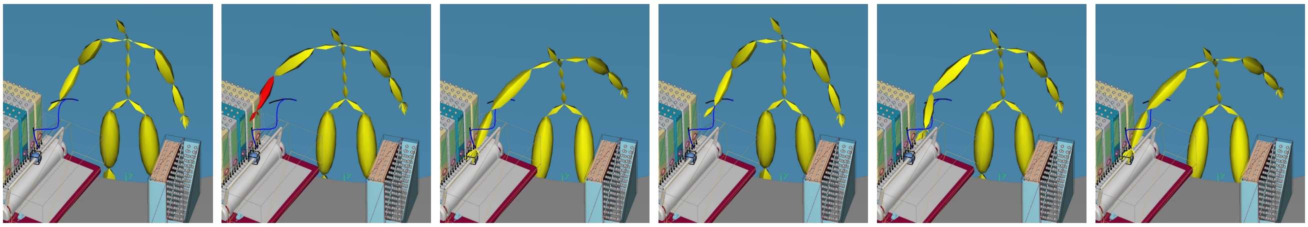

Fig. 20. Action synthesis corrected by force fields in blending space. The left three pictures show an action that produced collisions with obstacles; and the next three pictures show a collision avoidance solution (following the blue trajectory) for removing the collision.