Task-based Locomotion

Shailen Agrawal Michiel van de Panne*

University of British Columbia

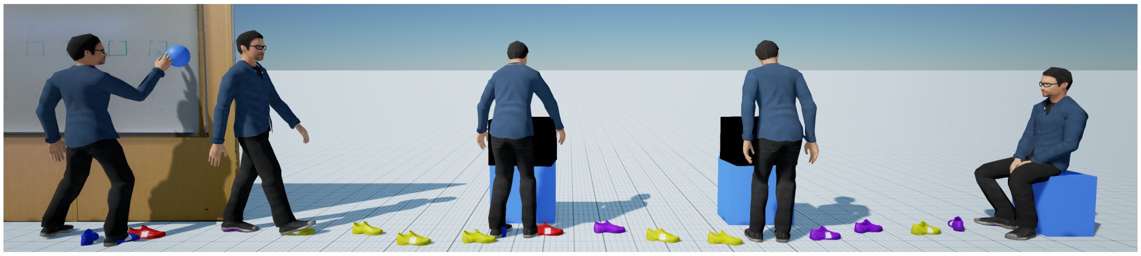


Figure 1: *Task-specific locomotion involving writing on a whiteboard, moving a box, and sitting on a box. The motion exhibits side-stepping, heel pivots, foot pivots, turns, and steps.*

Abstract

High quality locomotion is key to achieving believable character animation, but is often modeled as a generic stepping motion between two locations. In practice, locomotion often has task-specific characteristics and can exhibit a rich vocabulary of step types, including side steps, toe pivots, heel pivots, and intentional foot slides. We develop a model for such types of behaviors, based on task-specific foot-step plans that act as motion templates. The footstep plans are invoked and optimized at interactive rates and then serve as the basis for producing full body motion. We demonstrate the production of high-quality motions for three tasks: whiteboard writing, moving boxes, and sitting behaviors. The model enables retargeting to characters of varying proportions by yielding motion plans that are appropriately tailored to these proportions. We also show how the task effort or duration can be taken into account, yielding coarticulation behaviors.

Keywords: human locomotion, motion capture

Concepts: •**Computing methodologies** → **Animation; Physical simulation;**

1 Introduction

Animated human locomotion is an integral component of games, virtual reality, films, and training simulations. Believable movements accentuate immersion for all these applications. Locomotion is most often animated with the help of motion capture data and is usually abstracted as a sequence of alternate left-and-right footsteps. Clean footsteps, i.e., ones that are free of sliding and twisting motions on the ground during the contact phase, are often easier to work with, leading to the use of post-processing in order to remove footskate. This allows families of motion-captured steps to be interpolated and blended with each other with relative ease.

In practice, many motions also involve "messy" small-scale stepping movements that are used to reposition or reorient the body in preparation for a given task, or during an ongoing task. These steps may involve pivots, intentional foot-slides, side-steps, and partial steps and turns. A primary goal of our work is to be able to generate these types of nuanced motions for the task-specific contexts in which they occur. The locomotion steps used to carry a heavy box to a specific location are often different than the steps used to arrive at the same location in order to push a button. Our work is a pragmatic first step towards modeling the rich context-specific details that exist in many locomotion behaviors. We also aim to model coarticulation effects, where the planning of the locomotion steps in support of the current task can be influenced by the subsequent task. For example, a task requiring a brief action on the right followed by a longer duration task to the left may use only a partial side-step to the right to be within reach for the task, e.g., retrieving an object, and then proceeding with steps to the left.

Our method develops a footstep plan with the help of context-specific example footstep templates. The footstep locations and orientations derived from these templates are then further refined using online optimization. The simplicity of this approach avoids the complexities and large data requirements that are commonly required by statistical modeling approaches. Our task-based locomotion prototype is implemented in a modern game engine (Unreal Engine 4) and is demonstrated on three tasks: writing on a white board, pickup and placement of boxes, and sitting-down and standing-up. Figure 1 shows an example sequence of tasks and a visualization of the footstep plan that is key to our method.

The contributions of this paper are: (1) the identification of features that limit the realism of many generic locomotion methods, i.e., the lack of task-specific behavior, sometimes simplified foot-step vocabularies, and a limited ability to model coarticulation effects; (2) the development of a practical real-time solution with sparse data requirements that tackles these issues, using task-specific footstep templates coupled with online optimization.

*shailen|van@cs.ubc.ca

SIGGRAPH '16 Technical Paper,, July 24 - 28, 2016, Anaheim, CA,
ISBN: 978-1-4503-4279-7/16/07
DOI: http://dx.doi.org/10.1145/2897824.2925893

2 Related Work

A large body of literature explores the use of kinematic models for generating realistic human motion, most often with the help of motion capture data. Here we review only the most relevant work for our problem and proposed solution.

Motion sequencing: Many kinematic methods resequence existing motion data to create novel animations. A *motion graph* can be used to model the allowable ways in which motion capture clips can be sequenced while meeting given constraints on the quality of the transitions between clips. Full bodied character animations are then created by generating walks on these graphs [Kovar et al. 2002; Lee et al. 2002; Beaudoin et al. 2008; Yamane and Sok 2010; Min and Chai 2012], searching the interpolated motion graphs [Safonova and Hodgins 2007], building probabilistic models of transitions and character poses [Chai and Hodgins 2007; Wei et al. 2011], behavior-based planning algorithms [Lau and Kuffner 2005], or hierarchical controllers [Feng et al. 2012]. A common feature of these methods is that they discard the task-specific context in which the motion data has been captured. While it is still often possible to generate motions that are well matched to their context, this either happens serendipitously, or it needs to emerge from the use of an objective function or in response to constraints from the environment, i.e., side-stepping through a narrow gap. The task-specific foot-step templates at the heart of our method explicitly capture the contextual aspects of body positioning strategies, whereas motion graphs need to find ways to resythesize appropriate movements from motion clips that have been stripped of their context.

Motion embedding: In contrast to the above work, the context of a motion can be fully preserved by embedding motions directly in their environment. Prior work on *motion patches* [Lee et al. 2006] accomplished this via a mix of *motion tiles*. Smooth warping at the boundary transitions allows for a suitable degree of connectivity to be achieved between adjacent motion tiles. This model comes with its own limitations, namely that crossing a motion-patch or motion-tile boundary results in a discrete change of context. It is not obvious how to define tile sizes that are appropriate for many of the small-scale body-repositioning stepping motions whose nuances we seek to reproduce.

Footstep planning: Locomotion is often modeled in terms of the sequence of footstep locations, their orientations, and their timing. The foot placements should avoid obstacles, should follow natural stepping patterns, and make progress towards the desired goal. Footstep planning has been used as the basis for a variety of locomotion synthesis algorithms, e.g., [Van De Panne 1997; Choi et al. 2003; van Basten et al. 2011]. However motion planning that incorporates the rich stepping structure arising in a variety of natural task-specific contexts remains an open problem. Footstep planning has also been investigated for humanoid robots [Chestnutt et al. 2005; Kuffner et al. 2001; Kuffner et al. 2003]. Human locomotion often exhibits a rich stepping vocabulary, including side-steps, foot pivots, and intentional foot sliding. We believe that these substeps are a core component of believable character interactions with the environment.

Pose and motion reconstruction: Computing the skeletal joint parameters from end effector constraints can be computationally expensive. Multiple approaches have been proposed to alleviate some complexity, e.g., [Nakamura and Hanafusa 1986; Tolani et al. 2000; Aristidou and Lasenby 2011]. Inverse kinematics applied to characters as a whole is often informally called *full body IK*. When applied to human-like characters, full body IK can often produce unnatural solutions because most IK problems are under-constrained. Data-driven approaches can favor solutions which lie closer to the poses found in the motion database resulting in natural-looking poses [Grochow et al. 2004]. Similarly, statistical dynamical models, e.g., [Wang et al. 2007; Chai and Hodgins 2007] can be used to develop full motion sequences, including maximum-likelihood solutions that satisfy given spatio-temporal constraints, e.g., [Chai and Hodgins 2007]. In our work we use a *data-driven prior* to warm start an iterative IK algorithm [Aristidou and Lasenby 2011]. This produces natural-looking poses while still having the simplicity of an iterative IK algorithm.

Motion styles: A number of methods attempt to factor motions into independent attributes that might include motion attributes, e.g., walking speed, and style attributes, e.g., relating to the emotional state or other unique characteristics of a person's walking gait. Style generalization has been investigated for locomotion and other types of choreography, e.g., [Brand and Hertzmann 2000; Wang et al. 2007; Min et al. 2010]. Our proposed method provides a degree of support for motion styles as is implicit in the set of motion capture examples that serve as underlying templates.

Interacting with the environment: Realistic movement during interactions with the environment greatly adds to the believability of synthesized motions. This includes everyday tasks such as moving objects between shelves and tables, opening doors or closing doors and walking through them, and cooking in a kitchen. Realistic stepping for such scenarios remains an underexplored area, with a number of exceptions, e.g., [Yamane et al. 2004; Chai and Hodgins 2007; Safonova and Hodgins 2007; Min and Chai 2012]. We find inspiration in the work of [Yamane et al. 2004] towards synthesizing motions for moving objects between bookshelves, where one of our aims is to replace their static foot placements with natural foot stepping patterns. Of particular relevance to the problem of coarticulation is prior work on planning concurrent manipulations [Bai et al. 2012], which captures how humans exploit different properties of body parts and objects for multitasking. Our work is largely complementary, showing how the body position can be adapted in an efficient and natural fashion over time to support a given motion plan.

Reinforcement learning: Methods based on reinforcement learning (RL) have proved useful in developing highly capable behaviors for many tasks, including locomotion, e.g., [Lo and Zwicker 2008; Treuille et al. 2007; Lee and Lee 2006; Levine et al. 2012]. Despite impressive results, the motions are typically developed from motion clips that are then resequenced for the efficient generic navigation tasks, without regard for matching the context in which they were captured. They usually avoid close contact with the environment, and the motions are usually displayed on low-fidelity models that do not support detailed scrutiny, particularly of the feet. In principle, it may be possible to develop a suitable objective function that allows these methods to precisely produce all the nuances of human foot-stepping behaviors in different contexts. This has not been demonstrated in practice, in part because of the complexities of developing such an all-encompassing objective function. However, recent efforts in this direction have demonstrated good potential for finding objective functions that capture the overall path and body orientations favored by humans when walking from one position-and-orientation to another [Mombaur et al. 2010].

3 Overview

Given a sequence of task types and task locations as input, our system synthesizes the full-body animation required to move between the tasks using natural, task-specific locomotion patterns. Figure 2 shows an overview of our system, which we now review in detail.

Our primary goal is to develop high-quality motions for transitioning between *tasks*, where a task is defined as a location and orientation in the world where the character needs to accomplish some-

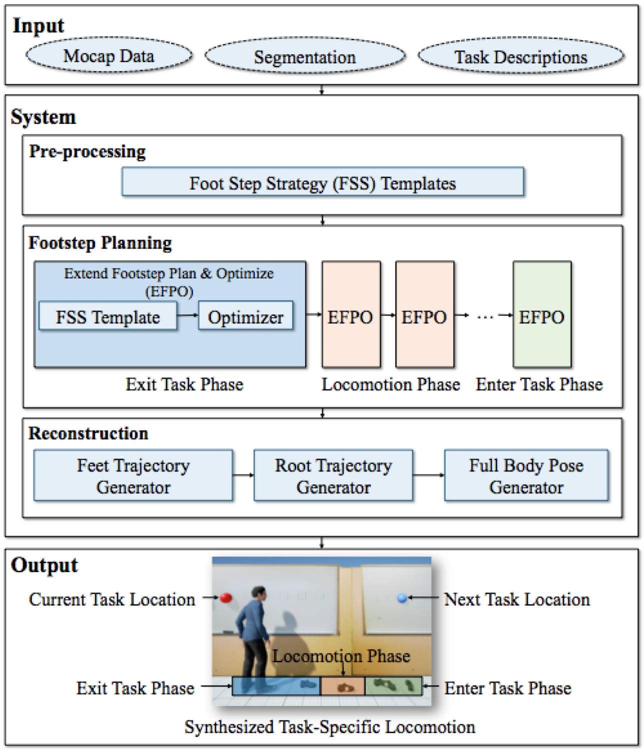
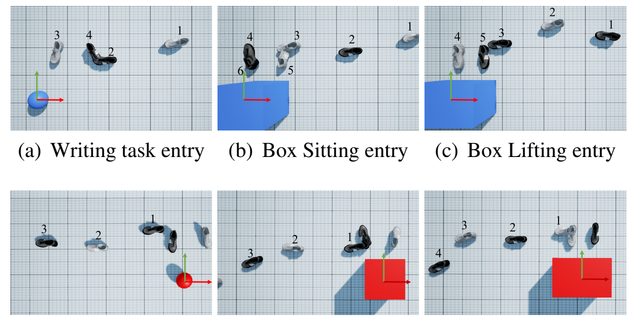


Figure 2: *System Overview.*

thing. We focus on four types of tasks, including writing on a board, moving boxes between locations, sitting on a box, and turning around in a specific location. In Figure 2(bottom), the task locations are marked using small spheres. A typical locomotion transition involves a pair of tasks, namely moving from a current-task location to a next-task location, marked with red and blue spheres, respectively. In addition to the task type and task location, we shall also later define and use a task *effort* attribute. This will allow for coarticulation effects to be modeled, such as choosing to temporarily step-and-reach towards a task location instead of taking further steps to place the body directly in front of a task location.

Footstep planning lies at the heart of our locomotion model, with the plan consisting of footstep locations and orientations that are to be achieved at the end of each step. As seen in the footstep planning component of Figure 2, the footstep plan is developed successively over several motion phases. Specifically, the transition between task locations is modeled as an *exit task* phase, a *locomotion* phase, and an *enter task* phase. Transitions between a pair of distant task locations will use all three of these phases, as shown in Figure 4(a). Shorter transitions may pass directly from *exit task* to *enter task*, or, if sufficiently close, directly to the *enter task* phase, as shown in Figure 4(b) and (c), respectively.

For each successive motion phase, an initial footstep plan is created by instantiating one of the template plans from a template library, which is specific to the current task type and the current motion phase. The template library is developed from the example motion capture data as an offline pre-processing step. The selection of the most suitable template is based on the quality of fit for a given template to the requirements of the current task, as will be described in the subsequent section. The footstep plan from the instantiated template is then optimized to satisfy a footstep-based objective function, which in general terms aims to reach a goal location while remaining close to the underlying template example



(a) Writing task entry (b) Box Sitting entry (c) Box Lifting entry

(d) Writing task exit (e) Box Sitting exit (f) Box Lifting exit

Figure 3: *Footstep plans for various task entries and exits.*

and satisfying smoothness criteria.

Given the optimized foot step plan, we then generate specific foot and root trajectories. Since each foot step in the template is associated with a specific segment of motion capture data, we use the associated feet and root trajectories and apply a smooth spatial warp in order to exactly achieve the given motion plan. Lastly, we generate full body motion from the reconstructed foot step and root trajectories, and the task description. Specifically, we begin with poses extracted from the motion segment associated with footstep template and then use full-body inverse kinematics as applied to the root, hands, and feet in order to reconstruct final poses that satisfy the desired task constraints.

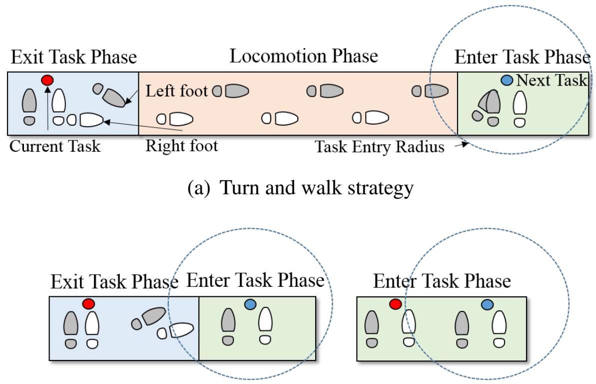
A more detailed summary of our method is given later in Algorithm 1. We will refer to specific steps in this algorithm by line number in the remainder of the paper, as needed. We now move on to providing more details on the core steps of our method.

4 Template-based Footstep Plans

A key aspect of our method is the use of example data to model the pattern of foot steps to be used during each motion phase. We will refer to these as *foot step strategy (FSS)* templates. Each FSS template further consists of individual steps. In this section we describe in detail how motion phases and steps are defined, and how a suitable FSS template is retrieved as the first step of planning the foot steps for any given motion phase. Importantly, the FSS templates are task-specific. For example, the templates used for task *entry* and *exit* for writing on a white board are markedly different from those for sitting on a box, picking up a box, or placing a box as illustrated in Figure 3 and detailed in Table 1.

4.1 Phases, Steps, and Templates

Motion Phases: Given a task transition, as described by a *current* task and a *next* task, the transition motion is modeled using three motion phases: task *exit*, *locomotion*, and task *entry*, as illustrated in Figure 4. Not all phases need to exist in any given synthesized transition motion. If one of the starting foot locations already lies within the *enter radius* for the next task, then only an entry phase is used, as shown in Figure 4 (c). This typically results in a side-stepping strategy. Otherwise, an exit phase is first planned, using an appropriately selected FSS template. If one of the planned steps resulting from that template passes within the enter radius, the enter phase is deemed to begin at that point in time, as seen in Figure 4 (b), and commonly results in a partial-turn-and-step strategy. Most typically, however, an additional locomotion phase is needed to plan foot steps from the end of the exit phase until one of the



(a) Turn and walk strategy

(b) Partial turn and step strategy (c) Side step strategy

Figure 4: *Motion phases and stepping strategies for several instances of writing-task transitions.*

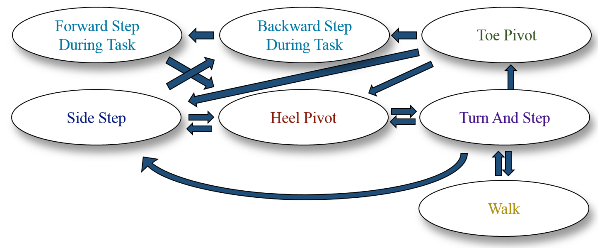


Figure 5: *A state diagram showing possible transitions between various footstep styles. The color coding used here for each footstep style is used in the results in the rest of the paper and in the video.*

planned locomotion steps lies within the enter task radius. The motion phases are generated in sequence, i.e., exit, locomotion, and entry, with an optimization step (§5) being applied after the template instantiation step for each of these motion phases.

Step segmentation: The foot-step templates are constructed from example motion data (§7) which is first segmented into individual steps. Each step starts when either the swing foot loses firm contact with the ground or it enters a sliding motion. A step ends when the swing foot re-establishes firm contact with the ground or it comes to rest for the case when it is undergoing a sliding motion. We categorize the observed foot steps as belonging to the following categories: Heel Pivot, Toe Pivot, Side Step, Turn and Step, Walk, Forward Step During Task and Backward Step During Task. The detailed interconnectivity of this rich step vocabulary is illustrated in Figure 5. The segmentation process is semi-automated: an initial automated pass is followed by a manual check and cleanup as necessary.

A step is modeled using a tuple

$$q = (n_s, n_e, n_{sa}, n_{ea}, \text{tag}, \text{phase}) \tag{1}$$

where n_s is the start frame of the step, n_e is the end frame of the step, n_{sa} is the start frame of the airborne portion of the step, n_{ea} is the end frame of the airborne portion of the step, tag is the category of the footstep (see Figure 5), and phase is task phase (*exit*, *locomotion*, or *enter*) of the segment. For a sliding motion such as a toe pivot or a heel pivot, n_{sa} and n_{ea} are set to be the same as n_s and n_e respectively as no distinction is made between airborne and sliding motion phases. We denote the set of all steps using

$$Q = \{q_i\}. \tag{2}$$

Table 1: *Foot Step Strategies*

FSS (Entries & Exits)	FootStep Categories
Write entry (Fig. 3(a))	Walk (1), Walk(2), TurnAndStep(3), ToePivot(4)
Sit entry (Fig. 3(b))	Walk (1), Walk(2), Walk(3), TurnAndStep(4), ToePivot(5), ToePivot(6)
Lift entry (Fig. 3(c))	Walk (1), Walk(2), Walk(3), TurnAndStep(4), TurnAndStep(5)
Write Exit (Fig. 3(d))	HeelPivot (1), TurnAndStep(2), Walk(3)
Sit Exit (Fig. 3(e))	HeelPivot (1), TurnAndStep(2), Walk(3)
Lift Exit (Fig. 3(f))	HeelPivot (1), TurnAndStep(2), Walk(3), Walk(4)

Motion templates: Sequences of steps from the example motions require assignment to the three motion phases, i.e., *exit*, *locomotion*, and *entry*. This is accomplished by first identifying the steps belonging to the *locomotion* phase using a simple-but-effective heuristic, namely that the yaw angle of the swing foot relative to the forward axis of the root frame falls below a given threshold. Foot steps taken before this are tagged as belonging to the *exit* phase and those after belong to the *entry* phase. For examples that never use a locomotion phase, steps are evenly split between the exit and enter phases if the number of footsteps exceed an empirically defined threshold (4 footsteps). For foot step patterns less than this threshold, all the foot steps are deemed to belong to the enter phase of the target task.

An exit template is defined as:

$$s^{\text{exit}} = (q^{\text{exit}}_{1:N}, \Gamma) \tag{3}$$

where $q_{1:N}$ is a sequence of steps that begin with an exit phase, $q^{\text{exit}}_{1:N}$ is $\{q_i | q_i \in q_{1:N} \wedge q_i(\text{phase}) = \text{exit}\}$, and Γ is the task transition descriptor which will be described shortly. Similarly, an entry template is defined as:

$$s^{\text{entry}} = (q^{\text{entry}}_{1:N}, \Gamma) \tag{4}$$

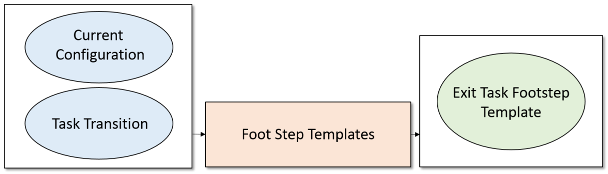
where, $q^{\text{entry}}_{1:N}$ is $\{q_i | q_i \in q_{1:N} \wedge q_i(\text{phase}) = \text{enter}\}$, and Γ is the task transition descriptor.

A *task transition* descriptor is modeled as a tuple of the form:

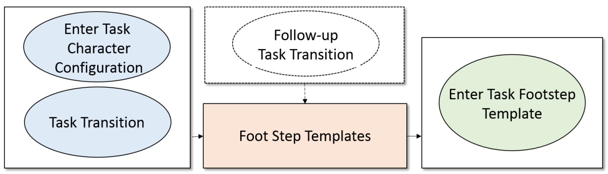
$$\Gamma = (c, \mathbf{x}^c_n, \mathbf{x}^n_c, \alpha_c, \alpha_n) \tag{5}$$

where c is the task category, x_c and x_n are the respective world-frame locations of the current and next tasks, $\mathbf{x}^n_c$ is the start location of the current task in the coordinate frame of the next task, $\mathbf{x}^c_n$ is the end location of the next task in the coordinate frame of the current task, α_c is the effort required for the current task, and α_n is the effort required for the next task, where the use of this *effort* attribute is described in what follows below.

Motion coarticulation: Low effort tasks exhibit a high degree of *coarticulation* with any follow-up tasks, meaning that the followup motion influences how the low effort task is executed. An example task that exhibits coarticulation for our work is that of a brief tap at a specified location on a whiteboard followed by a higher effort writing task, i.e., longer duration, also at a pre-specified location. When a low effort task is on the way towards performing a high effort task, the low effort task can be performed “in passing”. On

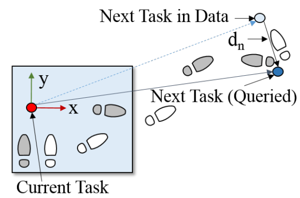


(a) Task Exit

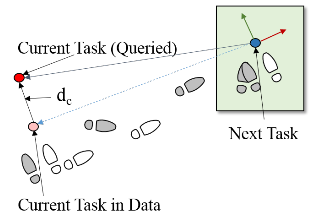


(b) Task Entry

Figure 6: *Reconstructing task-specific footstep templates. The most suitable template is selected for the given task phase based on similarity of the transition.*



(a) Exit Template Selection



(b) Entry Template Selection

Figure 7: *Task-exit and task-entry template selection.*

the other hand, if a low effort task lies in the opposite direction of a high effort task, then the observed motion consists of leaning to complete the low-effort task and then moving on to complete the high effort task. Successfully modeling this kind of coarticulation increases the realism of task-specific locomotion, and is neglected in existing methods. Our framework accounts for coarticulation during template selection. We model the degree of effort required for a task by classifying a task as a low effort or a high effort task. We use α to represent effort requirement for a task, taking on a value of 0 or 1 for low effort and high effort tasks, respectively. We note that our notion of *effort* is a qualitative one that captures the need to dwell in a comfortable position directly in front of a task position as opposed to the alternative of simply being within reach.

4.2 Template Retrieval

At runtime, suitable templates need to be found for any given task transition. In the simplest scenario, a suitable *exit* template is found by looking for the template that is most similar to the current required transition, Γ. Here, similarity implies a similar task effort value and relative location of the *next* task, as measured in the coordinate frame which has its origin at the *current* task for a particular task category. Analogously, a suitable *entry* template is found by looking for the example template that best matches the required task effort and relative current location of the character as measured in the coordinate frame which has its origin at the *next* task for a required task category. Figure 7 illustrates these ideas, which capture the key elements of the most basic case, where a character is in double stance during the actual tasks, and is effectively paused at the desired location to execute the task. In this example, the desired task transition occurs between two writing tasks. Note that while similarity-of-location of the next task is important for template selection, the *next* category type does not need to match when choosing the exit template.

Special cases: Beyond the common case just described, there are a number of scenarios that require the choice for task entry and the following task exit to be coupled together. For example, certain tasks such as turn-around (§7) always end in single stance, which then needs to be matched for the start of the next task. The entry template for a low effort task, such as a writing low effort task, might end with a single stance. We show an entry and exit template, in Figure 8(a), while planning for two successive transitions. In these cases there is an additional requirement for matching the last foot step of the entry template with the first footstep of exit template. For example, an entry template ending with a left stance should be followed by a right stance foot step in the exit template to be compatible, and vice versa. Hence, for such cases it is not sufficient to just choose an entry or exit template using the most suitable template according the similarity of the required transition Γ. Instead, we recover the k most suitable templates for entry and exit phases for use in template selection (Eqs. (6) and (7)).

$$\gamma^{\mathrm{exit}}(\Gamma_q^{\mathrm{exit}}) = s_{1:k}^{\mathrm{exit}} \tag{6}$$

$$\gamma^{\mathrm{entry}}(\Gamma_q^{\mathrm{entry}}) = s_{1:k}^{\mathrm{entry}} \tag{7}$$

We can then choose a combination of compatible entry and exit templates. This is done by minimizing the sum of distances for exit and entry templates with respect to the respective task descriptors using the distance metric defined in Eqs. 8 and 9. An example is shown in Figure 8(b) where same colored boxes indicate compatible footstep strategies for entry and exit across a task. Entry strategy 2) and Exit strategy 1) are chosen as the final pair because they give the least sum of distance metrics for a compatible pair of footstep strategies. We use $k = 5$ in all our experiments.

Template distance metric: A distance metric is used to rank the suitability of templates from the library with respect to the queried transitions. In order to achieve this we partition the exit and entry task descriptors for all the entry and exit task templates, respectively, using a kd tree. This allows for efficient retrieval of the footstep templates from the database which have task transition similar to the queried task transition. We use the following distance metric for recovering the k-most suitable templates:

$$D_{\mathrm{exit}}(\Gamma, \Gamma') = w_d d_n + w_e |\alpha_c - {\alpha_c}'| \tag{8}$$

where $d_n = ||\mathbf{x}_n^c - {\mathbf{x}_n^c}'||$, and $\mathbf{x}_n^c$ and ${\mathbf{x}_n^c}'$ define the next task location as seen in the coordinate frame of the current task, for the query case and template library instance, respectively. We use $w_d = 0.001$ and $w_e = 0.999$.

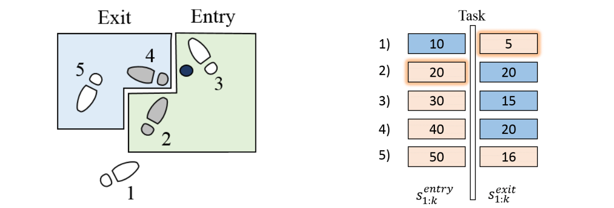
Similarly, the entry-template distance metric is defined by:

$$D_{\mathrm{entry}}(\Gamma, \Gamma') = w_d d_c + w_e |\alpha_n - {\alpha_n}'| \tag{9}$$

where $d_c = ||\mathbf{x}_c^n - {\mathbf{x}_c^n}'||$, and $\mathbf{x}_c^n$ and ${\mathbf{x}_c^n}'$ define the current task location as seen in the coordinate frame of the next task, for the query case and template library instance, respectively.

An example of the desired impact of D_{exit} in template selection is evident in Figures 4(a) and 4(b), where significantly different footstep strategies are used for these two cases.

In order to accomodate low effort tasks that exhibit coarticulation with follow-up tasks, we use a compound template and distance



(a) Task entry for turn-around ends with a single stance. (b) Template Combination Selection.

Figure 8: *Choice of the best pair of templates for entry and exit from k-most suitable footstep templates.*

metric that takes into account the current task transition as well as the followup task transition, as shown in Figure 6(b). The template distance metric when planning for coarticulation is defined by:

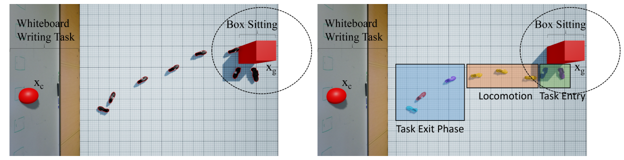
$$D^c_{\text{entry}}(\Gamma, \Gamma') = w_d d_c + w_f d_f + w_e(|\alpha_n - \alpha'_n| + |\alpha_f - \alpha'_f|) \quad (10)$$

where d_f is $||\mathbf{x}^n_f - \mathbf{x}^{n\prime}_f||$, $\mathbf{x}^n_f$ is the location of the followup task as measured in the coordinate frame of the next task, $\mathbf{x}^{n\prime}_f$ is the corresponding vector for the followup task in the template library, α_f is the effort required for the followup task, ${\alpha_f}'$ is the corresponding value for effort requirement of the followup task from the template library, and $w_f = 0.0005$ is the weight for the distance term corresponding to the followup task.

The use of D^c_{entry} for task entry allows for the generation of motions that can correctly reconstruct coarticulated motions as seen in the example data. For example, as seen in Figure 15(a) (§7), reconstructing motion for low effort tasks can perform a task 'in passing' when a low effort task is on the way towards performing a high effort task. Similarly, for a scenario where a low effort task lies on the left of the character followed by a high effort task on the right, the reconstructed motion uses a short side step, while leaning over to the left, in order to perform the quick low effort task, as shown in Figure 15(b) before proceeding to the high effort task on the right.

Footstep planning overview: Having detailed the template selection mechanism, we now describe the full footstep planning process. Using the desired task transition (Γ), we select the most suitable footstep strategy template for the task exit as per Eqs. (6) and (7) as described in Lines 7 and 13 of Algorithm 1. Using the current character configuration and the selected template we then generate a footstep plan for the exit task phase. Next, the locomotion phase is planned. The locomotion task phase uses footstep segments that are tagged with the "Walk" category. The first segment used for locomotion is selected so as to have a step length that is as similar as possible to that of the last foot step in the exit task phase, if a turn and step exists. Otherwise, we use a "preferred" left or right footstep motion segment for generating a locomotion footstep plan (Line 8). Planning for the locomotion phase is terminated when the enter-task criterion is satisfied, i.e., one of the planned locomotion foot step lies within the enter-task radius of the next task. Finally, we select the most suitable footstep strategy template using Eq. (4) and the distance metric define in Eqs. (8), (9) and (10) for the enter task phase (Lines 9 and 22).

While the footstep plan invoked by the templates provides a good starting point, further optimization is required for several reasons. The template-based footstep plan is limited in its ability to produce "good" footsteps, as shown in Figure 9(a), due to the use of a sparse template library. Also, transitions between exit, locomotion and entry phases require special care in order to generate natural foot



(a) Unoptimized footstep plan (b) Optimized footstep plan

Figure 9: *Comparison of unoptimized task-specific footstep plan with an optimized plan. The character is directed to exit from a writing task on the left represented by the sphere and sit on a box located on the right.*

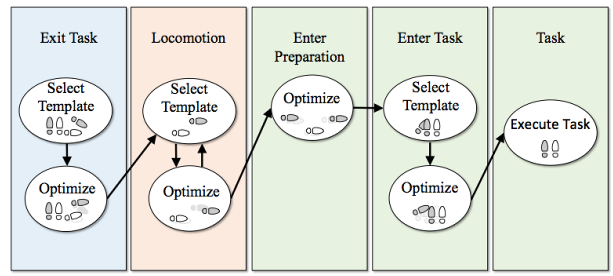


Figure 10: *Optimization phases.*

stepping behaviors. Please refer to the supplementary video for an example of a motion reconstructed from an unoptimized footstep plan. Hence, we need to further optimize the initial template-based footstep plan in order to produce a high-quality footstep plan, as shown in Figure 9(b).

5 Optimization

An online optimization procedure is used to adapt the template-based footstep plan in order to achieve a desired high-quality motion that precisely matches the desired heading direction, has smooth step-length variation, and so forth. The positions and orientations of the footsteps are the free variables in the optimization. We perform the optimization in several phases, as shown in Figure 10. The use of the task-aware and phase-specific motion templates together with the subsequent optimization allows for the desired generalization of footstep plans. We optimize for data prior, step smoothness, heading orientation, distance from goal and average local step location objectives, according to the following objective function:

$$f = w_d * f_d + w_s * f_s + w_o * f_o + w_{dg} * f_{dg} + w_l * f_l \quad (11)$$

We use BFGS as the optimization algorithm and, accordingly, develop the required analytic gradients for this method. We now describe each of the objective function terms in further detail.

5.1 Data Prior Objective

To produce a footstep plan that satisfies task constraints while remaining as close to the data as possible, we use the following term which penalizes deviation from the data:

$$f_d = \sum_{i=1}^{n}((x_i - x_{i_d})^2 + w_a * (y_i - y_{i_d})^2 + w_\theta * (\theta_i - \theta^d_i)^2) \quad (12)$$

where, x_i and y_i are the ground-plane components of the vector describing the next step in relation to the current step, x_{i_d} and y_{i_d}

are the same components as seen for the template data, $w_a = 2$ is a weight that further penalizes the step length component, and $w_\theta = 0.4$ weights the differences in footstep orientations.

5.2 Smooth Step Objective

It is desirable to have smooth variations in the generated footstep lengths as humans show a preference for minimizing energy expenditure by minimizing accelerations. The term encouraging slow changes to step lengths is defined by:

$$f_s = \sum_{i=1}^{n} (l_i - l_{i-1})^2 \quad (13)$$

where, l_i and l_{i-1} are the step lengths of i^{th} and $(i-1)^{\text{th}}$ footstep respectively.

5.3 Average Foot Orientation Objective

Locomotion is generally comprised of individual footsteps which match the overall heading direction. However, for task-specific templates, the original structure of the template should be preserved. This can be achieved by keeping the contribution of a desired-orientation objective small initially, and making it larger with each subsequent footstep in the plan. Using this approach, the optimized template will progressively orient towards the desired heading direction as the strategy executes, i.e., footsteps at the beginning of the optimized template maintain orientations similar to those of the original template whereas the later footsteps gradually show a preference for the desired heading direction. This is encapsulated with an objective function term as follows:

$$f_o = \sum_{i=1}^{n} ({\alpha_o}^{n-i} * (\theta_i - \theta_g)^2) \quad (14)$$

where θ_i is the orientation of the i_{th} footstep, θ_g is the relative orientation of the next task relative to the orientation of the first footstep in the current template. We use an $\alpha_o = 0.9$ in all our experiments.

5.4 Distance From Goal Objective

In order to create smooth transitions between the locomotion and enter task phases, we introduce the *Enter Preparation* phase (see Figure 10), which we shall describe in further detail below. This adapts the locations of the last few ($n = 3$) footsteps before the *enter* task phase begins. We want the last locomotion footstep to be placed at the best location for executing the footstep plan of the subsequent enter-task phase. This is performed by penalizing the distance of the footsteps of interest from the optimal footstep locations as defined by the subsequent enter task footstep strategy template. The objective term that helps achieve this is defined as follows:

$$f_{dg} = \sum_{i=1}^{n} (\alpha_{dg}^{n-i} * {d_i}^2) \quad (15)$$

where d_i is the distance between the i^{th} footstep and the footstep preceding the first footstep of the enter task footstep strategy template as recovered from the database, and $\alpha_{dg} = 0.9$ is a weighting factor.

5.5 Average Foot Location Objective

During the optimization process we want to minimize deviation from the straight-line path of the current template, as defined by the line segment connecting the first footstep of the current template and the goal location. We penalize the distances (d_i) of the average footstep locations for each pair of footsteps in sequence, from the line segment connecting the first footstep and the goal location according to following objective:

$$f_l = \sum_{i}^{n} d_i^2 \quad (16)$$

Enter Task Preparation : Transitioning from locomotion to the enter task phase requires some preparation close to the end of the locomotion phase. During planning for locomotion, when one of the planned foot steps satisfy entry criteria for transitioning to enter task phase, we select the most suitable template for the enter task. Next, the enter task footsteps are planned in the local coordinate frame of the desired next task. In order to optimally plan the location of the last locomotion footstep with respect to the first enter task footstep, additional locomotion foot steps may be introduced. For example, if the enter task template begins with a right swing footstep, and the locomotion phase also terminates with a right swing foot step, we introduce an additional left swing footstep in the planned locomotion footsteps. Additionally, if the transition footstep length between enter task footsteps and the last locomotion footstep is larger than a threshold, two additional locomotion footsteps are introduced to allow for footsteps of reasonable footstep length.

The last few footsteps ($n = 3$) before the enter task define an *Enter Preparation* phase (Figure 10). The footsteps belonging to this phase are re-optimized using the previously defined objective function, producing a locomotion footstep plan that appropriately prepares for the footstep strategy template associated with upcoming enter task.

During each motion phase, the various objectives have different relative importance. Thus, the optimization uses different weights for each objective function term during different phases of the planning algorithm. The weights we have used are empirically determined and are identical for all the results described in this paper. No additional tuning of these weights is required for use with various task categories. We tabulate the weights used for each phase in Table 2.

6 Full Body Motion Generation

The optimized footstep plan allows for relatively simple and efficient methods to be used for full body motion reconstruction. The reconstruction is performed in several phases as shown in Figure 2. First, foot trajectories are reconstructed using the synthesized footstep plan and the motion segments that are associated with each of these footsteps. Second, root motion trajectories are produced by warping the associated root motion trajectories. Finally, the foot and root trajectories, together with the task constraints, are

Table 2: *Weights for objective functions.*

Objective Function	Exit Task	Locomotion	Enter Task Prep	Enter Task
Data Prior	0.4	0.4	0.4	0.4
Smooth Step	0.1	0.2	0.1	0.1
Avg. Orientation	0.4	0.2	0.1	0.0
Distance From Goal	0.0	0.0	0.3	0.5
Avg. Location	0.1	0.2	0.1	0.0

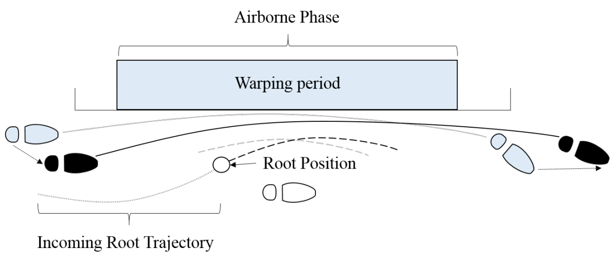


Figure 11: *Warping of a foot trajectory. Grey footsteps and curves represent original swing foot trajectory. Black footsteps and curves represent warped swing foot trajectory. The corresponding root motion is also warped using a transformation of the swing foot warp, along with an additional transformation to match the incoming root trajectory.*

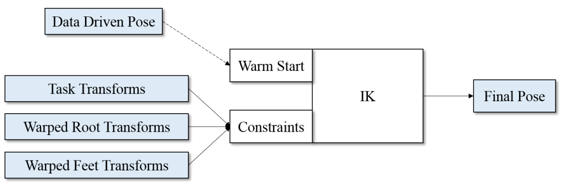


Figure 12: *The IK system takes feet, root and task related transforms as end effectors. The algorithm is initialized with a pose from the example motion data.*

then used to drive the end-effectors motion and a full-body inverse-kinematics (IK) procedure creates the final full-bodied motion.

We first describe the reconstruction of foot trajectories from the footstep plan. Each footstep in the synthesized plan has an associated task phase, category (heel pivot, toe pivot, side step, turn and step, forward/backward step or walking step) and an associated motion segment from the library of templates. We spatially warp the swing foot trajectory from the associated motion segment to the start and end locations, and orientations as specified by the task-specific footstep plan (Figure 11). A footstep can either be sliding or have an airborne component. For footsteps that have an associated air borne phase, we only perform the warping during the air-borne phase. However, since sliding footsteps do not have an associated airborne phase, they can be warped for the entire duration of the footstep. Since the synthesized footstep plan has been optimized with respect to a number of objectives, such as smooth step variation and data preservation, natural footstepping motion is preserved via the aforementioned warping technique. The timing of the footstep plan is preserved from the original template. We have experimented with including time warping but did not observe improved motion quality. As fewer templates are used, time warping would likely begin to yield some benefits.

The warping of foot trajectories to match the footstep plan should also result in the root motion being adapted in an appropriate fashion. We apply half the swing foot warp to the root motion in order to compensate for the swing foot motion warping. We also use root motion warping to match incoming root motion trajectory in order to create smooth transitions between successive segments of the reconstructed motions. When combined with feet trajectory warping, this creates believable transitions between the synthesized footsteps. The warping for the foot and root trajectories is accomplished using linear interpolation of the required translational offset as a function of the fraction of the total distance traveled by the foot from start-to-end, and analogously for the rotational component of

Input : List of task descriptors (location, category, effort)
Data: Motion capture clips segmented into steps tagged with one of the labels : Heel Pivot, Toe Pivot, Side Step, Turn and Step, Walk, Forward Step During Task and Backward Step During Task
Output: Set of footstep location and orientations, associated motion segments, full body motion

Algorithm `TaskSpecificLocomotion()`
- Initialize a library of "Footstep Strategy Templates" for task exits and task entries from the following information for each example task pair :
  - Sequence of motion segments exiting current example task and entering next example task
  - Associated task descriptor for exit and enter task
- Compute task entry radii for each task category.
- **foreach** *Task pair T_{i-1} and T_i* **do**
  - **known** : Foot plants from executing task T_{i-1}.
  - Compute task description vectors for queried exit and enter task.
  - `PlanTaskExitPhase()`
  - `PlanLocomotionPhase()`
  - `PlanTaskEntryPhase()`
  - Generate warped feet motion and root motion using the footstep plan and associated motion segments.
  - Generate full body motion using poses from the associated motion segments and full-body inverse kinematics as applied to the root, hands and feet.
- **end**

Procedure `PlanTaskExitPhase()`
- Select exit task footstep strategy template.
- `Optimize()`

Procedure `PlanLocomotionPhase()`
- **repeat**
  - Generate locomotion step.
  - `Optimize()`
- **until** *Until within task radius of T_i*
- Re-optimize last three footsteps to better match enter task footstep requirements.

Procedure `PlanTaskEntryPhase()`
- Select enter task footstep strategy template.
- `Optimize()`

Procedure `Optimize()`
- Optimize the newly generated template-based footsteps for the current phase with respect to the following objectives:
  - Data Prior
  - Smooth step length variation between consecutive steps
  - Orientation towards the task
  - Minimize distance from goal as the plan progresses
  - Average feet location to lie on the line between start and goal

Algorithm 1: A high level description of the task-specific motion planning algorithm.

the transformation matrices. The feet and the root transforms along with task constraints are used as end effectors for an iterative IK algorithm that has support for joint constraints and arbitrary joint hierarchies [Aristidou and Lasenby 2011]. The IK algorithm is initialized with a pose that is sampled from our database of poses associated with the current footstep and task constraint (Figure 12).

7 Results

We now describe the results obtained using our algorithm. The quality of the animated results is best seen in the video that accompanies this paper. Motion capture data is collected using 8 Vi-

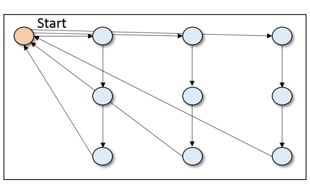
Table 3: *Description of task categories. The motion capture setups are described in Figure 13.*

Task Category	Motion Capture Setup	Data Size in mins.
Writing	a,b,c	3
Move Boxes	b,c	1.5
Box Sitting	b	0.85
Turn Around	b	N/A

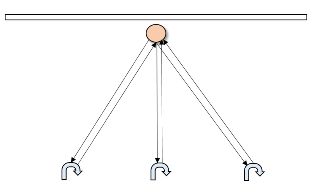
con MX40 [vic] cameras at 60 Hz, which is then down-sampled to 30 Hz for use in our system. We use 53 markers for capturing the actor while performing various tasks along with 4 markers to capture objects such as the box the actor was required to lift and move. The weight of the box in our example was 10 kg.

We explore four task categories in our work: writing on a whiteboard, moving boxes, sitting on a box, and turning around. The motion capture data captures the characteristic footstep strategies exhibited while entering and exiting tasks. We show a representation of the various motion capture setups used for collecting the relevant data in Figure 13. The starting point for a task is depicted by an orange circle, while each of the next task location is shown by blue circles or circular arrows. The arrows pointing between task locations represent the transitions between the respective task. Capturing data for each of the task categories utilizes one or more of these motion capture setups. We describe the task configurations for the data collected using the motion capture setups depicted in Figure 13 for each of the task categories in Table 3. We now describe each of the task categories in more detail.

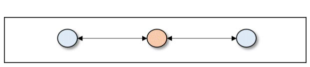
Writing : We use writing as a task as it captures footstep strategies that are characteristic of many similar everyday tasks. The effort is controlled by specifying the duration or the style of writing to be performed. Data is collected for writing tasks using the setups shown in Figures 13(a) and 13(b), with the goal of being able to reconstruct exit and entry strategies from any angle for a writing task. In order to collect data for a high effort form of the writing task, motion capture was performed for a high effort task where the actor was asked to draw several circles at pre-specified locations on the whiteboard using the setup shown in Figure 13(a). This figure depicts a front view of the task setup. Here the actor starts off by writing at the location specified by the first circle on the left in the figure, and then progressively writing in each of the subsequent columns followed by returning to performing a writing task at the location of the starting circle. Another set of data was recorded using the setup shown in Figure 13(b) which shows a top view of setup used. Here, the circle represents the actual task location, and



(a) Multiple column task (Front View).

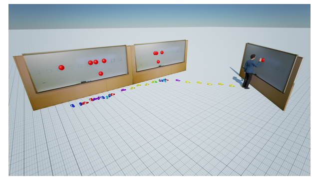


(b) Multiple angles of entries and exits (Top View).

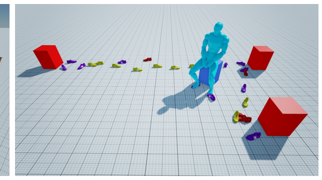


(c) Sideways task (Top View).

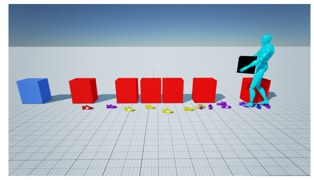
Figure 13: *Representative setups for recording various entry and exit strategies.*



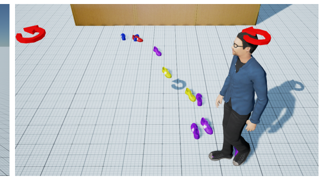
(a) Writing task.



(b) Box Sitting task.



(c) Box Moving.



(d) Turn Around Task.

Figure 14: *Task Categories.*

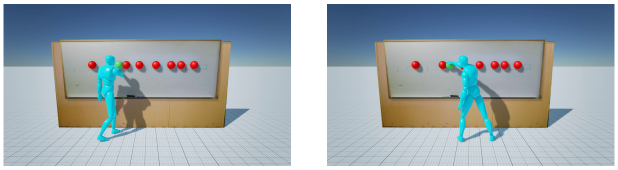
the arrows represent turnaround locations. We also recover footstep strategies associated with turning around from this setup as specified later in this section.

For capturing the interplay between low and high effort tasks, the actor was instructed to either tap within a box (low effort), or draw several circles (high effort) in a pre-specified sequence. The task setup shown in Figure 13(c) was used for recording this data. The actor was asked to start performing the task at the first specified location followed by tasks specified at the subsequent locations. Hence, the actor was always aware of both the next and the follow-up tasks while performing the current task. This allowed for the capture of co-articulation behavior which can arise while performing low effort tasks. In total, we collect nearly 3 minutes of motion captured data for the writing task. An example of a result generated for a writing task is shown in Figure 14(a)

Box Sitting: Sitting on a box requires unique foot step strategies that are markedly different from those observed in a writing task, both while entering and exiting the task. For example, for a sitting task, when approaching the target, the actor must first turn around before entering a sitting position. Hence, these unique footstep strategies are modeled as their own task category within our task-specific framework. All sitting tasks are considered to belong to a high effort category because they cannot be easily coarticulated with any other motion goals. We use 50 seconds of motion capture data for this task. We show an example of reconstructed footstep plan and generated pose for this task in Figure 14(b)

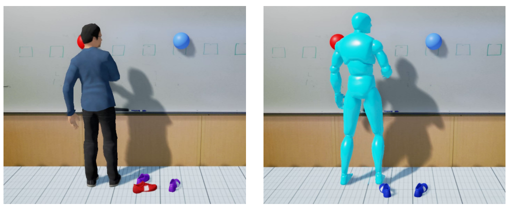
Moving a Box: Tasks such as lifting and moving boxes also have unique associated footstep strategies and we categorize this as a "high effort" task as it is also difficult to coarticulate with other followup tasks. Foot slides and foot pivots are rarely present while performing this task. The knowledge of the context in which these tasks are captured allows for reconstructing these associated characteristics using our framework. We use 1.5 minutes of motion capture data for representing various footstep strategies associated with lifting and moving boxes tasks. The weight of the box the actor was required to move was 10 Kg. The motion capture setups shown in Figure 13(b) and Figure 13(c) were used for this task. We show an example of an optimized footstep plan for this task in Figure 14(c)

Turn Around: A task-specific framework can also be used to perform a motion task such as turn-around. Footstep strategies for turn-around motions were collected as a byproduct as the relevant data can be recovered from the data pertaining to other tasks where a turning around motion is involved. For example, footstep strate-



(a) Approach from the left (b) Approach from the right

Figure 15: *A demonstration of coarticulation for two different approaches to an initial low-effort task, with the followup task being located to the right.*



(a) Shorter Character (b) Taller Character

Figure 16: *Effect of character anthropometry of synthesized task-specific foot step plan.*

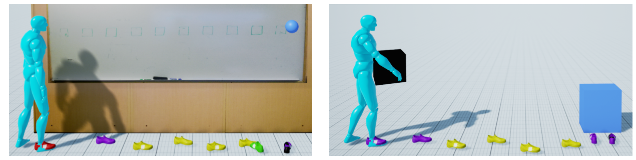
gies for the turn-around task are captured when a setup shown in Figure 13(b) is used for a writing task or a box sitting task. In Figure 14(d), the character is shown executing a turn-around task where the character exits from a writing task on the left of the whiteboard and walks towards the turn-around location, represented by a circular arrow, in order to execute it.

Task effort: Examples of motion coarticulations arising from low-effort tasks are illustrated in Figure 15. In Figure 15(a), the character reaches for a target while on the way to another target further to the right. In Figure 15(b), the character begins by performing a task to the right of the low-effort next task, and therefore performs a small left side step while leaning left to execute the low effort task, followed by turning and walking towards the next task on the right.

Effect of anthropometry: Our system generates footstep plans which are commensurate with the anthropometry of the character being used for the motion planning. For example, for the same task description, a taller character (Figure 16(b)) uses a side step instead of a partial turn and step as seen for the shorter character (Figure 16(a)). This difference in footstep plans arises from the use of character-specific footstep templates, which are generated by first motion retargeting the input animations for each character. We use MotionBuilder to perform the motion retargeting.

Ground-truth comparison: As a benchmark to compare against, we capture a motion sequence involving two writing tasks, a sit task, and a box-movement task. This is then treated as leave-out data, i.e., the footstep patterns are not used as templates. We then synthesize a motion for this sequence of tasks using our system. The ground-truth and synthesized motions can be seen in the video associated with this submission. We believe that this type of Turing test will continue to be useful in the future as motion synthesis methods continue to improve in their capabilities.

Impact of number of templates and task type: In the video we illustrate how the synthesized *exit* and *entry* motions degrade when the number of templates is reduced for the turn-around and writing tasks. We halve the number of templates used for the shown examples. Figure 17 shows how the task-entry footstep plan differs



(a) Writing task (b) Box-placement task

Figure 17: *Effect of the task on the synthesized foot-step plan.*

for a writing and a box-placement task that each start and end in the same location. In particular, the box lifting lacks the heel pivots and toe pivots that are found in the writing task.

8 Conclusions

In this paper we have proposed a model and algorithms for task-specific locomotion for arbitrary sequences of these tasks. Tasks are specified by choosing the desired location, orientation, category and effort of task to be performed. The synthesized footstep patterns and the resulting full-body motion faithfully reconstructs many of the features and observed strategies in a task-specific fashion. Hence, the proposed system allows for a directable approach for task-specific motion styles. The use task-specific locomotion models allows for the synthesis of realistic character motion in order to complement the ever-increasing visual realism of characters. By using an enhanced footstep vocabulary, the proposed model aims to *preserve* naturally occurring footstep features such as foot pivots, small foot shifts, and foot sliding, in a task-aware fashion rather than eliminating them, as is often the case for current motion synthesis techniques. The model also naturally models the the natural transitions between side-stepping, partial-turn-and-steps, and full-turn-and-step behaviors that occur in moving between tasks at varying distances. The synthesis of motion co-articulation effects serves to further improve realism.

Our work has a number of limitations. The footstep plans are described principally by their location and orientation. The current approach may not scale well to tasks that require higher-dimensional characterization. The current method has no explicit model of attention, and this can be noticable. Some aspects of the current algorithm still involve the use of heuristics. Human motions are not perfectly repeatable, unlike the output from our method. The results are therefore still dependent on the specific motion data used to construct the footstep templates.

We wish to further generalize the approach by creating parameterized task categories. For example, tasks related to object placement could be parameterized according to their force and precision requirements, or speed-of-task-completion requirements. Such task abstraction would allow for more general reuse of the footstep templates. We plan to explore integration with motion planners that implement collision avoidance as well as collaborative motions with other characters. Nuanced stepping models should also be useful for generating crowd animations whose motions stand up to close scrutiny. Person-specific stepping styles could be modeled for specific tasks, or modeled across multiple tasks.

9 Acknowledgments

We thank the anonymous reviewers for their helpful feedback and NSERC for funding via a Discovery Grant (RGPIN-2015-04843).

References

ARISTIDOU, A., AND LASENBY, J. 2011. Fabrik: A fast, iterative solver for the inverse kinematics problem. *Graphical Models 73*, 5, 243 – 260.

BAI, Y., SIU, K., AND LIU, C. K. 2012. Synthesis of concurrent object manipulation tasks. *ACM Transactions on Graphics (TOG) 31*, 6, 156.

BEAUDOIN, P., COROS, S., VAN DE PANNE, M., AND POULIN, P. 2008. Motion-motif graphs. In *Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, Eurographics Association, Aire-la-Ville, Switzerland, Switzerland, SCA '08, 117–126.

BRAND, M., AND HERTZMANN, A. 2000. Style machines. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, SIGGRAPH '00, 183–192.

CHAI, J., AND HODGINS, J. 2007. Constraint-based motion optimization using a statistical dynamic model. *ACM Transactions on Graphics (TOG) 26*, 3, 8.

CHESTNUTT, J., LAU, M., CHEUNG, G., KUFFNER, J., HODGINS, J., AND KANADE, T. 2005. Footstep planning for the honda asimo humanoid. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, 629–634.

CHOI, M. G., LEE, J., AND SHIN, S. Y. 2003. Planning biped locomotion using motion capture data and probabilistic roadmaps. *ACM Transactions on Graphics (TOG) 22*, 2, 182–203.

FENG, A. W., XU, Y., AND SHAPIRO, A. 2012. An example-based motion synthesis technique for locomotion and object manipulation. In *Proceedings of the ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, 95–102.

GROCHOW, K., MARTIN, S. L., HERTZMANN, A., AND POPOVIĆ, Z. 2004. Style-based inverse kinematics. In *ACM SIGGRAPH 2004 Papers*, ACM, New York, NY, USA, SIGGRAPH '04, 522–531.

KOVAR, L., GLEICHER, M., AND PIGHIN, F. 2002. Motion graphs. *ACM Transactions on Graphics*, 473–482.

KUFFNER, J.J., J., NISHIWAKI, K., KAGAMI, S., INABA, M., AND INOUE, H. 2001. Footstep planning among obstacles for biped robots. In *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, vol. 1, 500–505 vol.1.

KUFFNER, J., KAGAMI, S., NISHIWAKI, K., INABA, M., AND INOUE, H. 2003. Online footstep planning for humanoid robots. In *Robotics and Automation, 2003. Proceedings. ICRA '03. IEEE International Conference on*, vol. 1, 932–937 vol.1.

LAU, M., AND KUFFNER, J. J. 2005. Behavior planning for character animation. In *Proceedings of the 2005 ACM SIGGRAPH/Eurographics symposium on Computer animation*, ACM, 271–280.

LEE, J., AND LEE, K. H. 2006. Precomputing avatar behavior from human motion data. *Graphical Models 68*, 2, 158–174.

LEE, J., CHAI, J., REITSMA, P. S., HODGINS, J. K., AND POLLARD, N. S. 2002. Interactive control of avatars animated with human motion data. In *ACM Transactions on Graphics (TOG)*, vol. 21, ACM, 491–500.

LEE, K. H., CHOI, M. G., AND LEE, J. 2006. Motion patches: building blocks for virtual environments annotated with motion data. In *ACM Transactions on Graphics (TOG)*, vol. 25, ACM, 898–906.

LEVINE, S., WANG, J. M., HARAUX, A., POPOVIĆ, Z., AND KOLTUN, V. 2012. Continuous character control with low-dimensional embeddings. *ACM Transactions on Graphics (TOG) 31*, 4, 28.

LO, W.-Y., AND ZWICKER, M. 2008. Real-time planning for parameterized human motion. In *Symposium on Computer Animation*, Eurographics Association, 29–38.

MIN, J., AND CHAI, J. 2012. Motion graphs++: A compact generative model for semantic motion analysis and synthesis. *ACM Trans. Graph. 31*, 6 (Nov.), 153:1–153:12.

MIN, J., LIU, H., AND CHAI, J. 2010. Synthesis and editing of personalized stylistic human motion. In *Proceedings of the 2010 ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, ACM, New York, NY, USA, I3D '10, 39–46.

MOMBAUR, K., TRUONG, A., AND LAUMOND, J.-P. 2010. From human to humanoid locomotionan inverse optimal control approach. *Autonomous robots 28*, 3, 369–383.

NAKAMURA, Y., AND HANAFUSA, H. 1986. Inverse kinematic solutions with singularity robustness for robot manipulator control. *Journal of dynamic systems, measurement, and control 108*, 3, 163–171.

SAFONOVA, A., AND HODGINS, J. K. 2007. Construction and optimal search of interpolated motion graphs. *ACM Trans. Graph. 26*, 3 (July).

TOLANI, D., GOSWAMI, A., AND BADLER, N. I. 2000. Real-time inverse kinematics techniques for anthropomorphic limbs. *Graphical Models 62*, 5, 353 – 388.

TREUILLE, A., LEE, Y., AND POPOVIĆ, Z. 2007. Near-optimal character animation with continuous control. *ACM Transactions on Graphics (TOG) 26*, 3.

VAN BASTEN, B. J. H., STÜVEL, S. A., AND EGGES, A. 2011. A hybrid interpolation scheme for footprint-driven walking synthesis. In *Proceedings of Graphics Interface 2011*, Canadian Human-Computer Communications Society, School of Computer Science, University of Waterloo, Waterloo, Ontario, Canada, GI '11, 9–16.

VAN DE PANNE, M. 1997. From footprints to animation. *Computer Graphics Forum 16*, 4, 211–223.

Vicon. http://www.vicon.com/. Accessed: 31-05-2015.

WANG, J., FLEET, D., AND HERTZMANN, A. 2007. Multifactor gaussian process models for style-content separation. In *Proceedings of the 24th international conference on Machine learning*, ACM, 975–982.

WEI, X., MIN, J., AND CHAI, J. 2011. Physically valid statistical models for human motion generation. *ACM Trans. Graph. 30*, 3 (May), 19:1–19:10.

YAMANE, K., AND SOK, K. 2010. Planning and synthesizing superhero motions. *Motion in Games*, 254–265.

YAMANE, K., KUFFNER, J. J., AND HODGINS, J. K. 2004. Synthesizing animations of human manipulation tasks. *ACM Trans. Graph. 23*, 3 (Aug.), 532–539.